

# **„Entwicklung eines roboterbasierten Prüfstands für die Ermittlung der Langzeitbetriebsfestigkeit von beweglichen Karosserieteilen“**

Von der Fakultät für Maschinenbau, Verfahrens- und Energietechnik  
der Technischen Universität Bergakademie Freiberg

genehmigte

## **DISSERTATION**

zur Erlangung des akademischen Grades

Doktor – Ingenieur

(Dr.-Ing.)

vorgelegt

von: Dipl.-Ing. (FH) Jens Vogelpohl  
geboren am: 25. Juli 1980 in Georgsmarienhütte

Gutachter:	Prof. Dr.-Ing. habil. Peter Löber	TU Bergakademie Freiberg
	Prof. Dr.-Ing. Martin Reike	HS Osnabrück

Tag der Verleihung: 22. Januar 2013

# I Danksagung

Diese Dissertation entstand nebenberuflich während meiner Tätigkeit als Versuchingenieur bei der Bertrand AG in der Niederlassung München. Diese Arbeit ist eine Fortführung meiner Diplomarbeit.

Herrn Prof. Dr.-Ing. habil. Peter Löber von der Technischen Universität Bergakademie Freiberg danke ich sehr herzlich für die Gelegenheit zur Promotion. Seine wohlwollende Unterstützung und die mir gewährten Freiheiten ermöglichten die Realisierung dieses Projekts in Kooperation mit der Hochschule Osnabrück. Herrn Prof. Dr.-Ing. Martin Reike von der Hochschule Osnabrück gilt mein herzlicher Dank für die Betreuung und Übernahme des Koreferats.

Ohne die Finanzierung meines Arbeitgebers wäre die Durchführung dieses Projekts nicht möglich gewesen. Für das mir geschenkte Vertrauen danke ich überaus dem Niederlassungsleiter Bernd Mogwitz, dem Niederlassungsleiter Dr. Gerrit Schmidt, dem Abteilungsleiter Karl – Heinz Weinberger sowie meinem Teamleiter Philip Class. Für wertvolle Tipps und praktischer sowie theoretischer Unterstützung danke ich den aktuellen und ehemaligen Kollegen Thomas Birner, Paul Bühler, Stefan Czekalla, Mathis Fichter, Christian Herweiger, Jürgen Kaiser, Klaus Keitel, Mario Kernbach, Marc – André Köppen, Florian Leitl, Julian Mesz, Michael Mondelli, Klaus Müller, Korbinian Oestreicher, Mathias Rietzler, Frank Roppelt, Pascal Sourisseau, David Weyer und Alexander Woithe. Anja Schauser danke ich für die Unterstützung bei der Veröffentlichung von Zeitschriftenartikeln.

Weiterhin gilt mein Dank Prof. Dr.-Ing. Dirk Rokossa von der Hochschule Osnabrück für die kritischen Hinweise und wertvollen Anregungen, Dr. Raimund Funke vom Institut für Umweltanalytik aus Möhrendorf für die uneigennützige Leihgabe eines Testroboters sowie Dr. Stefan Riesner von der Firma RTLeaders für die fördernden Informationen und Kontakte.

Meiner Freundin Katja und meinen Eltern Maria und Manfred danke ich für Rücksichtnahme, moralische Unterstützung und die kritische Durchsicht des Manuskripts.

Jens Vogelpohl

## II Inhaltsverzeichnis

<b>I</b>	<b><i>Danksagung</i></b> .....	<b>2</b>
<b>II</b>	<b><i>Inhaltsverzeichnis</i></b> .....	<b>3</b>
<b>III</b>	<b><i>Abkürzungsverzeichnis</i></b> .....	<b>8</b>
<b>IV</b>	<b><i>Nomenklatur / Formelzeichen</i></b> .....	<b>10</b>
<b>V</b>	<b><i>Abbildungsverzeichnis</i></b> .....	<b>12</b>
<b>VI</b>	<b><i>Tabellenverzeichnis</i></b> .....	<b>15</b>
<b>VII</b>	<b><i>Formelverzeichnis</i></b> .....	<b>16</b>
<b>1</b>	<b><i>Einleitung</i></b> .....	<b>17</b>
1.1	Problemstellung und Motivation .....	18
1.2	Zielsetzung .....	19
1.3	Vorgehensweise .....	20
<b>2</b>	<b><i>Reale und virtuelle Erprobung in der Fahrzeugentwicklung</i></b> .....	<b>22</b>
2.1	Geschichte der Fahrzeugerprobung .....	22
2.2	Virtuelle Entwicklung und Erprobung.....	25
2.3	Grenzen der Simulation .....	28
2.4	Fortschritt durch Symbiose beider Verfahren.....	30
<b>3</b>	<b><i>Stand der Technik und Forschung</i></b> .....	<b>32</b>
3.1	Prüfstandssysteme in der Automobilindustrie .....	32
3.1.1	Standardprüfstand .....	32
3.1.2	Einzellösung.....	33
3.2	Betriebsfestigkeitsuntersuchungen von Karosserieteilen .....	34
3.2.1	Voraussetzungen für die reale Erprobung .....	35
3.2.2	Potentielle Mängel am Prüfling .....	36
3.2.2.1	Konzept- und Konstruktionsfehler .....	36
3.2.2.2	Material- und Fertigungsfehler.....	37
3.2.2.3	Montagefehler .....	37
3.2.2.4	Softwarefehler .....	38
3.2.2.5	Sonstige Fehler .....	38

3.2.3	Prüfvorschriften .....	39
3.2.4	Prüfstände für Karosseriekomponenten.....	40
3.3	Robotik.....	41
3.3.1	Robotertypen.....	42
3.3.1.1	Industrieroboter .....	42
3.3.1.2	Serviceroboter .....	43
3.3.1.3	Humanoide Roboter .....	43
3.3.1.4	Visionen der Roboterforschung.....	44
3.3.2	Roboterwerkzeuge .....	44
3.3.3	Koordinatensysteme.....	45
3.3.4	Roboterprogrammierung.....	46
3.3.4.1	Online – Programmierung .....	47
3.3.4.1.1	Teach – In – Verfahren .....	47
3.3.4.1.2	Playback – Verfahren.....	48
3.3.4.2	Offline – Programmierung .....	48
3.3.4.2.1	Textbasierte Verfahren .....	48
3.3.4.2.2	Virtuelle Verfahren .....	49
3.3.4.3	Hybride Verfahren.....	49
3.3.4.4	Aufgabenorientierte Programmierung.....	50
3.3.5	Förderung der Einsatzmöglichkeiten für Industrieroboter .....	51
3.3.6	Mensch – Roboter – Kooperation.....	55
3.3.7	Zusammenfassung .....	56
3.4	Fahrzeugelektronik.....	56
3.4.1	Steuergeräte .....	57
3.4.2	Bussysteme .....	57
3.4.3	On – Board – Diagnose.....	58
3.4.3.1	ODX .....	59
3.4.4	Restbussimulation .....	60
<b>4</b>	<b><i>Ein Prüfsystemkonzept für den Komponentenversuch .....</i></b>	<b>62</b>
4.1	Anforderungen .....	62
4.1.1	Anforderungsliste.....	63
4.1.2	Erläuterung der Anforderungen .....	64
4.1.2.1	Energie.....	64
4.1.2.2	Ergonomie .....	64
4.1.2.3	Gebrauch .....	65



4.1.2.4	Geometrie und Kinematik .....	66
4.1.2.5	Kontrolle.....	67
4.1.2.6	Kosten.....	67
4.1.2.7	Kräfte.....	68
4.1.2.8	Sicherheit.....	68
4.1.2.9	Signale .....	68
4.1.2.10	Wartung .....	69
4.1.3	Umsetzung .....	70
4.2	Verwendete Software.....	70
4.2.1	LabVIEW.....	71
4.2.2	CANoe .....	72
4.3	Softwarebasierte Systemarchitektur .....	74
4.3.1	Datenbasis .....	75
4.3.2	Hard- und Softwareschnittstellen .....	77
4.3.2.1	Analoge Eingänge .....	78
4.3.2.2	Digitale Kanäle.....	78
4.3.2.3	Robotik .....	80
4.3.2.4	Batterie – Ladecomputer .....	80
4.3.2.5	Umweltsimulation .....	81
4.3.2.6	Mobiltelefon .....	81
4.3.2.7	CANoe.....	82
4.3.3	Parameterverwaltung .....	82
4.3.4	Auswertungskonfigurator .....	83
4.3.5	Ansteuerungskonfigurator .....	85
4.3.6	Messdatenverwaltung .....	86
4.3.7	Mess- und Analysetool .....	87
4.3.8	Ereignisanzeige .....	89
4.3.9	Prüfablaufverwaltung.....	90
4.3.10	Prüfmodule.....	92
4.3.11	Softwaretools für Einzelprüfungen.....	93
4.3.12	Hauptmenü .....	93
4.4	Messdatenerfassung und Analyse .....	94
4.4.1	Kontinuierliche Messdatenerfassung .....	94
4.4.2	Auswirkung variabler Umgebungsbedingungen .....	98
4.4.3	Überwachung zeitraffungsbedingter Effekte.....	100

4.5	Verwendete Hardware .....	102
4.5.1	Prüfstandsrechner.....	102
4.5.2	Robotik.....	103
4.5.2.1	Einsatz in der Komponentenerprobung.....	103
4.5.2.2	Vergleich von Robotermodellen .....	104
4.5.2.3	Arbeitssicherheit.....	108
4.5.2.4	Einsatz unter extremen Klimabedingungen .....	108
4.5.3	Spannungsversorgung.....	109
4.5.4	Umweltsimulation.....	110
4.5.5	Messmittel.....	110
4.5.5.1	Drehwinkelgeber .....	111
4.5.5.2	Näherungssensor .....	111
4.5.5.3	Stromsensor .....	112
4.5.5.4	Kraftsensor .....	113
4.5.5.5	Roboterbasierte Wegmessung .....	113
4.6	Aufgabenorientierte Schnittstelle zur Robotik .....	114
4.6.1	Anforderungen .....	114
4.6.2	Funktionsprinzip .....	116
4.6.3	Kommunikationsprotokoll.....	117
4.6.3.1	Steuerbotschaften .....	118
4.6.3.1.1	Position auslesen.....	118
4.6.3.1.2	Bedienerkommunikation.....	119
4.6.3.1.3	Koordinatensystem festlegen.....	119
4.6.3.1.4	Überschleifen.....	125
4.6.3.1.5	Bewegungsfreigabe.....	125
4.6.3.1.6	Messmittel- und Werkzeugparameter.....	126
4.6.3.1.7	Messdatenexport .....	127
4.6.3.1.8	Achsmomentexport.....	128
4.6.3.1.9	Armleistung und Betriebsart.....	129
4.6.3.2	Bewegungsbotschaften.....	130
4.6.3.2.1	Position anfahren .....	130
4.6.3.2.2	Achsenbewegungen .....	132
4.6.3.2.3	Linearbewegungen.....	133
4.6.3.2.4	Kreisbewegungen.....	135
4.6.4	Umsetzung auf dem Robotercontroller.....	137
4.6.5	Inbetriebnahme einfachster Betätigungen .....	139

4.6.6	Parametrierung komplexer Bewegungsabläufe .....	142
<b>5</b>	<b><i>Realisierung eines Prüfstands zur Dauererprobung von Fahrzeugtüren .....</i></b>	<b><i>147</i></b>
5.1	Prüfungsspezifische Hardware.....	147
5.1.1	Roboterarme.....	147
5.1.2	Roboterwerkzeuge .....	148
5.1.2.1	Roboterwerkzeug zur Türaußenbetätigung .....	149
5.1.2.2	Roboterwerkzeug zur Türinnenbetätigung.....	150
5.1.3	Adaption der Roboter am Fahrzeug.....	151
5.2	Prüfmodule zur Türerprobung .....	154
5.2.1	Türbewegung .....	155
5.2.2	Türschlossbetätigung .....	157
5.2.3	Fensterheberbetätigung.....	158
5.2.4	Diagnosefunktion.....	159
5.3	Aufbau, Inbetriebnahme und Durchführung.....	159
<b>6</b>	<b><i>Realisierung eines Prüfstands für Steifigkeitsmessungen.....</i></b>	<b><i>161</i></b>
6.1	Prüfstandsspezifische Hardware .....	162
6.2	Kompensation der Federsteifigkeit des Roboterarms.....	163
6.3	Bedienoberfläche .....	166
<b>7</b>	<b><i>Bewertung.....</i></b>	<b><i>169</i></b>
7.1	Prüfsystemkonzept.....	169
7.2	Einsatz in der Praxis.....	171
7.3	Investitionsbetrachtung .....	173
<b>8</b>	<b><i>Zusammenfassung und Ausblick .....</i></b>	<b><i>175</i></b>
<b>9</b>	<b><i>Thesen.....</i></b>	<b><i>178</i></b>
<b>10</b>	<b><i>Quellenverzeichnis .....</i></b>	<b><i>181</i></b>

### III Abkürzungsverzeichnis

6D	Sechsdimensional
ActiveX	Active Extensions
AG	Aktiengesellschaft
ASCII	American Standard Code for Information Interchange
AVL	Anstalt für Verbrennungskraftmaschinen List
A – Bot	Adaptive robot programming
Basic	Beginner's All – purpose Symbolic Instruction Code
CAD	Computer – aided design
CAE	Computer – aided engineering
CAN	Controller Area Network
CANoe	Bezeichnung einer Software der Firma Vector Informatik
CAPL	Communication Access Programming Language
CARS21	Competitive Automotive Regulatory System for the 21st century
CFK	Carbon – faserverstärkter Kunststoff
DaLaBert	Dauerlaufsoftware Bertrandt
DBC	CANdb – Datenbank – Datei
DBL	Deutronic Batterieladegerät
DIN EN ISO	Deutsches Institut für Normung Europäische Norm Internationale Organisation für Normung
DLL	Dynamic Link Library
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DMU	Digital Mock – Up
DOP	Data Objekt Property
DVD	Digital Versatile Disc
EIA	Electronic Industries Alliance
EVA	Eingabe – Verarbeitung – Ausgabe
F/W	Forderungen / Wünsche
FEA	Finite Elemente Analyse
FEM	Finite Elemente Methode
FDX	Fast Data Exchange
FH	Fensterheber
GSM	Global System for Mobile Communications
HIL	Hardware in the Loop
HL	Hinten links
HR	Hinten rechts
IBM	International Business Machines Corporation
INI	Umgangssprachlich für Näherungssensor
IP	Ingress Protection
IRoProg	Industrielle Roboter Programmierung
KFZ	Kraftfahrzeug
KK	Klimakammer
KS	Koordinatensystem

LabVIEW	Laboratory Virtual Instrumentation Engineering Workbench
LED	Lichtemittierende Diode
LIN	Local Interconnect Network
MKS	Mehrkörpersimulation
MCD	Measure Calibrate Diagnose
MELFA	Mitsubishi Electric Factory Automation
MORPHA	Intelligent Anthropomorphous Assistance Systems as a Basic Variant of Man – Machine – Interaction
MOST	Media Oriented Systems Transport
MvS	Monate vor Serienanlauf
NI	National Instruments
ODX	Open Diagnostic Data Exchange
OPC	Object Linking and Embedding for Process Control
OSIRIS	Ein objektorientiertes System zur impliziten Roboterprogrammierung und Simulation
PC	Personal Computer
PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect Express
PKW	Personenkraftwagen
PORTHOS	Portable Handhabungssysteme für den ortsflexiblen Einsatz in der Produktion
ProDemo	Modellbasierte Programmierung by Demonstration
PXI	Peripheral Component Interconnect Extensions for Instrumentation
RS – 232	Radio Sector – 232
RT	Real Time
SCARA	Selective Compliance Assembly Robot Arm
SMERobot	The European Robot Initiative for Strengthening the Competitiveness of Small and Medium Enterprises in Manufacturing
SMS	Short Message Service
SOP	Start of Produktion
SPS	Speicherprogrammierbare Steuerung
TCP/IP	Transmission Control Protocol / Internet Protocol
TCP	Tool Center Point
TDL	Türdauerlauf
TÜV	Technischer Überwachungsverein
UPD	User Datagram Protocol
USA	United States of America
USB	Universal Serial Bus
VAL3	Variable Assembly Language 3
VL	Vorne links
VR	Vorne rechts
VW	Volkswagen
Z3	Zuse Z3
ZE	Zentral entriegeln
ZS	Zentral sichern
ZV	Zentral verriegeln

## IV Nomenklatur / Formelzeichen

Symbol	Einheit	Beschreibung
$A$	—	Transformationsmatrix zur Synchronisation von Koordinatensystemen
$\vec{A}$	—	Hilfsvektor zur Berechnung des Türkoordinatensystems
$a$	$\frac{N}{mm^2}$	Amplitude der potentiellen Ausgleichsfunktion
$a_{11} \dots a_{33}$	—	Koeffizienten der Transformationsmatrix
$\vec{a}$	—	Richtungsvektor der Bauteildrehachse
$B$	—	Exponent der potentiellen Ausgleichsfunktion
$\vec{B}$	—	Hilfsvektor zur Berechnung des Türkoordinatensystems
$\vec{C}$	—	Hilfsvektor zur Berechnung des Türkoordinatensystems
$D$	—	Allgemeiner Richtungsvektor
$\vec{D}$	—	Hilfsvektor zur Berechnung des Türkoordinatensystems
$d$	mm	Abstand zum Ursprung des Roboterkoordinatensystems
$E_{XY}$	—	X – Y – Ebene des Roboterkoordinatensystems
$E_{XZ}$	—	X – Z – Ebene des Roboterkoordinatensystems
$E_{YZ}$	—	Y – Z – Ebene des Roboterkoordinatensystems
$F$	N	Kraft auf den Roboterarm
$f_1 \dots f_3$	—	Koeffizienten des Verschiebungsvektors
$\vec{f}$	—	Verschiebungsvektor zur Synchronisation von Koordinatensystemen
$\vec{g}_1, \vec{g}_2$	—	Hilfsgeraden zur Berechnung des Türkoordinatensystems
Hz	$s^{-1}$	Hertz
$\vec{P}_0$	mm	Lage des Ursprungs eines beliebigen Koordinatensystems
$\vec{P}_{0_{FzgKS}}$	—	Ursprung des Fahrzeugkoordinatensystems
$\vec{P}_1 \dots \vec{P}_4$	mm	Mit einer Messspitze angefahrene Punkte zur Berechnung der Lage von Koordinatensystemen
$\vec{P}_{1,r} \dots \vec{P}_{4,r}$	mm	Vermessene Punkte am realen Bauteil zur Synchronisation von Koordinatensystemen
$P_{1,r_x} \dots P_{4,r_z}$	mm	Koeffizienten der Punkte am realen Bauteil zur Synchronisation von Koordinatensystemen
$\vec{P}_{1,v} \dots \vec{P}_{4,v}$	mm	Exportierte Punkte aus dem CAD – Modell eines Bauteils zur Synchronisation von Koordinatensystemen
$P_{1,v_x} \dots P_{4,v_z}$	mm	Koeffizienten exportierter Punkte aus dem CAD – Modell eines Bauteils zur Synchronisation von Koordinatensystemen
$\vec{P}_{FzgKS}$	mm	Beliebiger Punkt im Fahrzeugkoordinatensystem im CAD – Modell
$\vec{P}_{RobKS}$	mm	Beliebiger Punkt im Roboterkoordinatensystem

$\overrightarrow{P_X}$	mm	Punkt auf der X – Achse eines beliebigen Koordinatensystems
$\overrightarrow{P_{X_{FzgsKS}}}$	mm	Punkt auf der X – Achse im Fahrzeugkoordinatensystem
$\overrightarrow{P_{XY}}$	mm	Punkt in der X – Y – Ebene eines beliebigen Koordinatensystems
$\overrightarrow{P_{XY_{FzgsKS}}}$	mm	Punkt in der X – Y – Ebene im Fahrzeugkoordinatensystem
$R_X$	°	Winkel um die X – Achse im dazugehörigen Koordinatensystem
$R_Y$	°	Winkel um die Y – Achse im dazugehörigen Koordinatensystem
$R_Z$	°	Winkel um die Z – Achse im dazugehörigen Koordinatensystem
$\vec{r}$	–	Richtung der linearen Roboterbewegung
$t$	s	Zeit
$S$	–	Schnittpunkt zweier Geraden
$W_d$	mm	Wegdifferenz
$W_{d_X}$	mm	Wegdifferenz in X – Richtung im Roboterkoordinatensystem
$W_{d_Y}$	mm	Wegdifferenz in Y – Richtung im Roboterkoordinatensystem
$W_{d_Z}$	mm	Wegdifferenz in Z – Richtung im Roboterkoordinatensystem
$W_e$	mm	Mit externer Sensorik gemessener Weg einer Roboterbewegung
$W_R$	mm	Mit robotereigener Sensorik gemessener Weg einer Roboterbewegung
$W_{R_K}$	mm	Korrigierte Messdaten des Wegs einer Roboterbewegung
$X$	mm	X – Koordinate im dazugehörigen Koordinatensystem
$x$	mm	X – Richtung im kartesischen Koordinatensystem
$Y$	mm	Y – Koordinate im dazugehörigen Koordinatensystem
$y$	mm	Y – Richtung im kartesischen Koordinatensystem
$Z$	mm	Z – Koordinate im dazugehörigen Koordinatensystem
$z$	mm	Z – Richtung im kartesischen Koordinatensystem
$\alpha$	rad	Winkelmaß
$\alpha_{XY}$	rad	Schnittwinkel mit der X – Y – Ebene im Roboterkoordinatensystem
$\alpha_{XZ}$	rad	Schnittwinkel mit der X – Z – Ebene im Roboterkoordinatensystem
$\alpha_{YZ}$	rad	Schnittwinkel mit der Y – Z – Ebene im Roboterkoordinatensystem
$\alpha'$	$\frac{\text{rad}}{\text{s}}$	Winkelgeschwindigkeit
$\alpha'_{\text{Min}}$	$\frac{\text{rad}}{\text{s}}$	Minimale Winkelgeschwindigkeit
$\lambda_1, \lambda_2$	–	Variablen der Geradengleichung
$\Delta t$	s	Schrittweite der numerischen Differenzierung
$\Delta t_{\text{Min}}$	s	Minimale Schrittweite der numerischen Differenzierung

## V Abbildungsverzeichnis

Abbildung 1: Fehlerbehebungskosten abhängig vom Abstellzeitpunkt [Lin-10] .....	18
Abbildung 2: Ausschnitte eines Werbefilms zum VW Käfer 1303 .....	23
Abbildung 3: Evolutionsstufen der Konstruktion [Her-07] .....	25
Abbildung 4: Zuordnung von Erprobungsschritten zu Entwicklungsphasen [Deu-06] .....	25
Abbildung 5: Tendenz zur Reduktion der Gesamtentwicklungszeit [Her-07] .....	26
Abbildung 6: Phasen im Entwicklungsprozess eines Fahrzeugs [Gla-09] .....	26
Abbildung 7: Simulation verschiedener Crashesituationen .....	28
Abbildung 8: Verbindung von realer und virtueller Erprobung [Boc-08] .....	30
Abbildung 9: Standardprüfstände .....	33
Abbildung 10: Einzellösungen .....	34
Abbildung 11: Dauerlaufeinrichtungen zur Türerprobung .....	40
Abbildung 12: Roboter in der Automobilproduktion [Bmw-11] .....	41
Abbildung 13: Roboterarm des Motoman SDA10D mit 15 Achsen [Mot-10] .....	42
Abbildung 14: Humanoider Roboter als Helfer auf der Baustelle [Boc-04] .....	44
Abbildung 15: Roboterhand des DLR [Gre-10] .....	45
Abbildung 16: Koordinatensysteme an einem Industrieroboter Roboter [Löb-07] .....	46
Abbildung 17: Programmierverfahren für Industrieroboter [Fri-10] .....	47
Abbildung 18: Aufbau eines Steuergeräts [Rei-08] .....	57
Abbildung 19: Anwendung der DOP zur Anzeige einer Temperatur [Zim-10] .....	59
Abbildung 20: Gemeinsamer Zugriff auf ODX – Daten [Wal-06] .....	60
Abbildung 21: Restbussimulation mit zwei realen und zwei simulierten Netzknuten [Göh-11] .....	61
Abbildung 22: Darstellung des Entwicklungsprozesses als V – Modell .....	70
Abbildung 23: Programmcode einer parallel ablaufenden Schleifenstruktur in LabVIEW .....	72
Abbildung 24: Bedienoberfläche von CANoe .....	73
Abbildung 25: Softwarebasierte Architektur des Prüfsystemkonzepts .....	74
Abbildung 26: Programmcode zur Zeitmessung des Zugriffs auf die Datenbasis .....	76
Abbildung 27: Auswahl verfügbarer Hardwareressourcen in der Komponente <i>Digitale Eingänge</i> .....	77
Abbildung 28: Manuelle und automatische Ansteuerung in der Komponente <i>Relais</i> .....	78
Abbildung 29: Winkelmaß und differenzierte Winkelgeschwindigkeit .....	79
Abbildung 30: Bedienoberfläche der Komponente <i>Batterie - Ladecomputer</i> .....	81
Abbildung 31: Bedienoberfläche der Komponente <i>Parameterverwaltung</i> .....	83
Abbildung 32: Bedienoberfläche zur Auswahl der Datenquellen zur Bestimmung des Prüflingsstatus <i>Türöffnung</i> .....	84
Abbildung 33: Programmcode zur Berechnung des Prüflingsstatus <i>Türöffnung</i> .....	84
Abbildung 34: Programmcode der Ansteuervariante <i>Direktbestromung Fensterhebermotor</i> .....	86
Abbildung 35: Bedienoberfläche der Komponente <i>Messdatenverwaltung</i> .....	87
Abbildung 36: Bedienoberfläche der Komponente <i>Mess- und Analysetool</i> .....	88



Abbildung 37: Bedienoberfläche der Komponente <i>Ereignisanzeige</i> und GSM – Mobiltelefon.....	89
Abbildung 38: <i>Prüfplaneditor</i> der Komponente <i>Prüfablaufverwaltung</i> .....	90
Abbildung 39: <i>Prüfablaufsteuerung</i> der Komponente <i>Prüfablaufverwaltung</i> .....	91
Abbildung 40: Interner Aufbau der Prüfmodule .....	92
Abbildung 41: Bedienoberfläche des <i>DaLaBert</i> Hauptmenüs .....	94
Abbildung 42: Erkennungssystem zur Verarbeitung von Messdaten [Rec-05] .....	95
Abbildung 43: Stromverlauf eines Fensterhebermotors .....	95
Abbildung 44: Verläufe der Maximalkraft und Betätigungsenergie bei einer Dauererprobung .....	96
Abbildung 45: Bedienoberfläche zur Konfiguration der Messdatenpräsentation .....	97
Abbildung 46: Bedienoberfläche zur Anzeige der Messdatenpräsentation.....	97
Abbildung 47: Fehler am Fensterheber im Verlauf einer Spannungsabfallsimulation .....	98
Abbildung 48: Farbskala zur Umgebungstemperatur.....	99
Abbildung 49: Bedienoberfläche der Messdatenpräsentation mit farblich gekennzeichneten Messdaten .....	99
Abbildung 50: Verlauf der Einklemmkräfte am Fensterheber bei einer Dauererprobung ....	100
Abbildung 51: Messdatenpräsentation mit Darstellung einer <i>Langen Pause</i> .....	101
Abbildung 52: Prüfstandsrechner .....	102
Abbildung 53: Spezialvorrichtung zur Klappenbetätigung .....	104
Abbildung 54: Robotersystem <i>Stäubli RX160L</i> mit Klimaschutzhülle .....	109
Abbildung 55: Versuchsaufbau einer Türerprobung in der Klimakammer.....	110
Abbildung 56: Drehwinkelgeber an der Beifahrertür.....	111
Abbildung 57: Näherungssensor an der Türinnenverkleidung.....	112
Abbildung 58: Vergleichsmessung zwischen Strommesszange <i>HZ56-2</i> und Stromsensor <i>HASS 50-S</i> .....	112
Abbildung 59: Kraftmesselement im Roboterwerkzeug .....	113
Abbildung 60: Messung des Kraftverlaufs beim Einclipsen eines Stifts in die Halterung ....	115
Abbildung 61: Hardwaretechnischer Aufbau der aufgabenorientierte Schnittstelle zur Robotik .....	116
Abbildung 62: Festlegen des Türkoordinatensystems.....	120
Abbildung 63: Punkte, Vektoren und Geraden zur Berechnung des Fahrzeugkoordinatensystems.....	122
Abbildung 64: Programmcode zur Berechnung der Lage der Punkte des Türkoordinatensystems.....	123
Abbildung 65: Punkte zur Berechnung der Lage des Fahrzeugkoordinatensystems .....	124
Abbildung 66: Programmcode zur Umrechnung vom Bauteil- ins Roboterkoordinatensystem .....	125
Abbildung 67: Gegenüberstellung von kartesischen Koordinaten und Kugelkoordinaten ....	131
Abbildung 68: Darstellung der Parameter zur Botschaft <i>Linearbewegung</i> .....	134
Abbildung 69: Darstellung der Parameter zur Botschaft <i>Kreisbewegung</i> .....	136
Abbildung 70: Tasks auf dem Robotercontroller .....	138
Abbildung 71: Mechanischer Aufbau zur Betätigung eines Gangwahlhebels .....	139
Abbildung 72: Prüfmodule zur Ausführung einfachster Bewegungen .....	140
Abbildung 73: Bedienoberfläche zur Parametrierung einer Kreisbewegung.....	140

Abbildung 74: Bedienoberfläche zur Konfiguration <i>einfachster Bewegungen</i> .....	141
Abbildung 75: Bedienoberfläche zur Steuerung einfachster Bewegungsabläufe .....	141
Abbildung 76: Bedienoberfläche der Messdatenanzeige .....	142
Abbildung 77: Bewegungsablauf des Roboterschritts <i>Tür entriegeln Roboter außen</i> .....	143
Abbildung 78: Bedienoberfläche zum Erzeugen einer Türbewegung.....	145
Abbildung 79: Bedienoberfläche zur Verwaltung der Parameter eines Grundgerüsts.....	146
Abbildung 80: Roboterwerkzeug zur Türaußenbetätigung .....	149
Abbildung 81: Roboterwerkzeug zur Türinnenbetätigung .....	150
Abbildung 82: Podest für den Roboter zur Türaußenbetätigung.....	151
Abbildung 83: Profilschiene zur Adaption der Roboter an Komplettfahrzeuge .....	152
Abbildung 84: Grundrahmen zur Adaption der Roboter an Rohkarosserien .....	152
Abbildung 85: Prüfaufbau zur Betätigung beider Türen an einer Fahrzeugseite.....	153
Abbildung 86: Gestell zur Adaption des Roboters im Fahrzeuginnenraum.....	154
Abbildung 87: Bilderserie des Prüfmoduls <i>Türbewegung</i> .....	155
Abbildung 88: Aufbau einer Prüfung zur Türerprobung.....	160
Abbildung 89: Prüfportal auf einer Messplatte mit elektrischem Servozyylinder .....	161
Abbildung 90: Prüfstand für Steifigkeitsmessungen .....	163
Abbildung 91: Versuchsaufbau zum Vergleich der Wegmessung zwischen roboterinterner und externer Sensorik .....	163
Abbildung 92: Vergleich der Wegmessung zwischen roboterinterner und externer Sensorik .....	164
Abbildung 93: Steigung der Wegdifferenz in Abhängigkeit des Abstands in X – Richtung.....	164
Abbildung 94: Programmcode zur Kompensation der Federsteifigkeit.....	165
Abbildung 95: Bedienoberfläche zur Ausführung der Vergleichsmessungen zur Kompensation der Federsteifigkeit des Roboterarms.....	167
Abbildung 96: Bedienoberfläche zur Positionierung des Prüfstempels .....	167
Abbildung 97: Bedienoberfläche zur Ausführung einer Steifigkeitsmessung .....	168
Abbildung 98: Bilderserie durchgeführter Türdauerläufe .....	171

## VI Tabellenverzeichnis

Tabelle 1: Zusammenfassung der Dissertation.....	17
Tabelle 2: Anforderungsliste .....	64
Tabelle 3: Datenquellen zur Bestimmung des Prüflingsstatus <i>Türöffnung</i> .....	83
Tabelle 4: Varianten zur Ansteuerung des Fensterhebers .....	85
Tabelle 5: Vergleich von Roboterarmen zur Türbetätigung.....	106
Tabelle 6: Zusammenfassender Vergleich der Roboterarme .....	107
Tabelle 7: Beispielhafte Syntax des Kommunikationsprotokolls .....	118
Tabelle 8: Syntax der Botschaft <i>Position auslesen</i> .....	119
Tabelle 9: Syntax der Botschaft <i>Bedienerkommunikation</i> .....	119
Tabelle 10: Syntax der Botschaft <i>Koordinatensystem festlegen</i> .....	121
Tabelle 11: Syntax der Botschaft <i>Überschleifen</i> .....	125
Tabelle 12: Syntax der Botschaft <i>Bewegungsfreigabe</i> .....	126
Tabelle 13: Syntax der Botschaft <i>Ladungsverstärker parametrisieren</i> .....	126
Tabelle 14: Syntax der Botschaft <i>Werkzeugparameter einstellen</i> .....	127
Tabelle 15: Syntax der Botschaft <i>Messdatenexport</i> .....	127
Tabelle 16: Syntax der Botschaft zur Übertragung der Messdaten .....	128
Tabelle 17: Syntax der Botschaft <i>Achsmomentexport</i> .....	128
Tabelle 18: Syntax der Botschaft zur Übertragung der Achsmomente .....	128
Tabelle 19: Syntax der Botschaft <i>Status Armleistung und Betriebsart</i> .....	129
Tabelle 20: Syntax der Botschaft <i>Armleistung ausschalten</i> .....	129
Tabelle 21: Syntax der Botschaft <i>Position anfahren</i> .....	131
Tabelle 22: Syntax der Botschaft <i>Achsenbewegung</i> .....	132
Tabelle 23: Syntax der Botschaft <i>Werkzeug ausrichten</i> .....	133
Tabelle 24: Syntax der Botschaft <i>Linearbewegung</i> .....	133
Tabelle 25: Syntax der Botschaft <i>Kraftbegrenzte Stoßbewegung</i> .....	134
Tabelle 26: Syntax der Botschaft <i>Parameterbasierte Kreisbewegung</i> .....	135
Tabelle 27: Syntax der Botschaft <i>Drehung im Türkoordinatensystem</i> .....	137
Tabelle 28: Grundgerüst des Roboterschritts <i>Tür entriegeln Roboter außen</i> .....	143
Tabelle 29: Grundgerüst zur Parametrierung des Roboterschritts <i>Tür entriegeln Roboter außen</i> .....	144
Tabelle 30: Ausgewählte Roboterarme zur Türbetätigung.....	148
Tabelle 31: Hauptfunktionsträger des Roboterwerkzeugs zur Türaußenbetätigung .....	149
Tabelle 32: Hauptfunktionsträger des Roboterwerkzeugs zur Türinnenbetätigung.....	150
Tabelle 33: Hauptfunktionsträger des Roboterpodests zur Türaußenbetätigung .....	151
Tabelle 34: Hauptfunktionsträger der Profilschiene für Komplettfahrzeuge .....	152
Tabelle 35: Interner Ablauf des Prüfmoduls <i>Türbewegung</i> .....	156
Tabelle 36: Interner Ablauf des Prüfmoduls <i>Türschlossbetätigung</i> .....	157
Tabelle 37: Interner Ablauf des Prüfmoduls <i>Fensterheberbetätigung</i> .....	158
Tabelle 38: Ausgewählter Roboterarm für Steifigkeitsmessungen .....	162
Tabelle 39: Materialkostenaufstellung .....	173

## VII Formelverzeichnis

Formel 1: Rückwärtsdifferenz zur numerischen Berechnung der Winkelgeschwindigkeit.....	79
Formel 2: Minimale Schrittweite zur numerischen Berechnung der Rückwärtsdifferenz .....	79
Formel 3: Richtungsvektor der Bauteildrehachse .....	121
Formel 4: Gerade 1 durch die Drehachse des Türkoordinatensystems .....	121
Formel 5: Gerade 2 durch die Drehachse des Türkoordinatensystems .....	122
Formel 6: Lambda 2 der Geradengleichung .....	122
Formel 7: Lineare Abbildungsgleichungen .....	123
Formel 8: Lineare Abbildung zur Synchronisation von Koordinatensystemen .....	124
Formel 9: Rotationswinkel $R_x$ aus kartesischen Koordinaten .....	131
Formel 10: Rotationswinkel $R_y$ aus kartesischen Koordinaten.....	132
Formel 11: Funktion der Wegdifferenz .....	165
Formel 12: Richtungsabhängige Wegdifferenz .....	165

# 1 Einleitung

Die Komplexität von Kraftfahrzeugen hat in den letzten Jahren beträchtlich zugenommen. Begründen lässt sich dies mit den gestiegenen Anforderungen an Komfort, Sicherheit sowie Umweltverträglichkeit und der damit verbundenen Zunahme von elektronischen Komponenten. Parallel dazu wächst die Variantenvielfalt zur Erschließung neuer Marktsegmente bei gleichzeitiger Verkürzung der Entwicklungszeiträume. Zur Erfüllung der Kundenwünsche sowie der betriebswirtschaftlichen Ziele der Automobilkonzerne sind permanent innovative Entwicklungen unverzichtbar. Für den Bereich der Validierung bedeutet dies eine Anpassung der angewandten Methoden an technische Innovationen und veränderte Rahmenbedingungen. Dieser Anspruch wurde im Jahr 2007 von den Versuchsabteilungen für Karosserieteile nicht zufriedenstellend erfüllt. Für Optimierungen an vorhandenen Prüfeinrichtungen erwies sich der Handlungsbedarf als zu umfangreich, um diesen im Rahmen von Studienabschlussarbeiten umzusetzen. Vielmehr war die grundlegende Planung und Umsetzung eines neuartigen und zeitgemäßen Prüfsystemkonzepts gefragt. Diese Dissertation beschreibt die Entwicklung eines software- und roboterbasierten Prüfsystemkonzepts bis zum erfolgreichen Einsatz in der industriellen Praxis. Tabelle 1 stellt die Gliederung dieser Arbeit zusammenfassend dar.

1 Einleitung	
2 Reale und virtuelle Erprobung in der Fahrzeugentwicklung	
3 Stand der Technik und Forschung	Prüfstandssysteme in der Automobilindustrie
	Betriebsfestigkeitsuntersuchungen von Karosserieteilen
	Robotik
	Fahrzeugelektronik
4 Ein Prüfsystemkonzept für den Komponentenversuch	Anforderungen
	Verwendete Software
	Softwarebasierte Systemarchitektur
	Messdatenerfassung und Analyse
	Verwendete Hardware
	Aufgabenorientierte Schnittstelle zur Robotik
5 Realisierung eines Prüfstands zur Dauererprobung von Fahrzeugtüren	Prüfungsspezifische Hardware
	Prüfmodule zur Türerprobung
	Aufbau, Inbetriebnahme und Durchführung
6 Realisierung eines Prüfstands für Steifigkeitsmessungen	Prüfstandsspezifische Hardware
	Kompensation der Federsteifigkeit des Roboterarms
	Bedienoberfläche
7 Bewertung	
8 Zusammenfassung und Ausblick	
9 Thesen	
10 Literatur	

**Tabelle 1: Zusammenfassung der Dissertation**

## 1.1 Problemstellung und Motivation

Die Automobiltechnik hat sich in den vergangenen 30 Jahren stark gewandelt. Anfang der 1980iger Jahre dominierte noch der Maschinenbau, 90 Prozent der auftretenden Probleme konnten mit einem Schraubendreher oder Ringschlüssel behoben werden. Heute erwarten die Kunden mit jeder neuen Fahrzeuggeneration einen Zuwachs an Komfort, Sicherheit, Zuverlässigkeit, Umweltfreundlichkeit und Sparsamkeit [Mül-06]. Hinzu kommen steigende gesetzliche Anforderungen, beispielsweise beim Emissionsausstoß oder Fußgängerschutz [Schm-07]. Die Folge ist ein stetiger Anstieg des Elektronikanteils am Gesamtfahrzeug. Derzeit basieren 90 Prozent der Innovationen im Automobil auf elektronischen Komponenten [Wal-06]. Mit dem Einzug der Mikroelektronik ins Kraftfahrzeug Anfang der 1990iger Jahre durch vernetzte Steuergeräte hat sich die Informatik als Teildisziplin der Automobilentwicklung etabliert [Bra-03].

Das moderne Automobil ist gegenwärtig der aufwendigste in Großserie produzierte Gegenstandsgegenstand. Aktuelle Fahrzeuge bestehen aus bis zu 20.000 Einzelteilen und erreichen damit einen äußerst hohen Komplexitätsgrad [Gös-01]. Diese Eigenschaft ist im Entwicklungsprozess während der Absicherung eines neuen Fahrzeugmodells zu berücksichtigen. Die Zunahme elektronischer Funktionen, die Verwendung von neuen Materialien und eine konstruktiv grenzwertige Auslegung als Resultat der verbesserten Berechnungsmethoden haben die Zahl der Fehlerquellen erhöht [Hau-04]. Die Identifizierung möglicher Fehler muss möglichst früh im Entwicklungsprozess erfolgen. Die Kosten zur Beseitigung von Mängeln steigen exponentiell, je später die Ursachen aufgedeckt werden [Nat-10a].

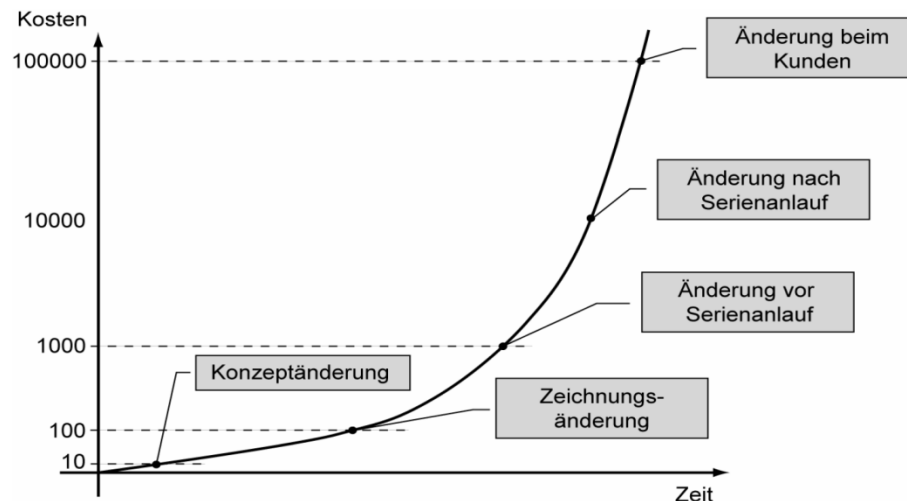


Abbildung 1: Fehlerbehebungskosten abhängig vom Abstellzeitpunkt [Lin-10]

Die Zuverlässigkeit eines Kraftfahrzeugs ist ein sehr wichtiges Kriterium beim Neuwagenkauf [Dat-10]. Die erwartete Laufleistung liegt bei mindesten 150.000 Kilometern in einem Einsatzzeitraum von 15 Jahren [Rei-08]. Zur Erfüllung dieser Erwartungen ist die stetige Anpassung der Prüftechnik an die Innovationen im Automobilbau notwendig, um damit zum Erhalt des Unternehmens beizutragen. Bei einigen Prüfstandstypen, beispielsweise Motoren- und Getriebepfstandstypen, ist dieser Gleichschritt von Produktentwicklung und Prüftechnik eine Selbstverständlichkeit. Andere Prüfstände und die damit eng verbundenen Prüfverfahren haben im letzten Jahrzehnt kaum eine Aktualisierung erfahren. Letzteres gilt für einen Großteil der Prüfstände im Bereich der Karosserieerprobung.

Meine Diplomarbeit *Automatisierung eines Türdauerlaufs mittels LabVIEW unter Zuhilfenahme der im Fahrzeug verbauten Elektronik und eines Knickarmroboters* [Vog-07a] erbrachte den Nachweis, dass der Einsatz von Robotik in der Komponentenerprobung prinzipiell realisierbar ist. Zusätzlich wurden Möglichkeiten aufgezeigt, wie durch eine Schnittstelle zur Fahrzeugelektronik kundennähere und hardwaretechnisch weniger aufwendige Prüfaufbauten zu verwirklichen sind. Somit war der erste Schritt zu mehr Realitätsnähe und zugleich zeit- sowie kostensparenden Prüfsystemen gemacht. Schnell zeigte der praktische Einsatz, dass trotz des offensichtlich vorhandenen Potentials für einen in der Praxis einsetzbaren Prüfstand noch einiges an Entwicklungsarbeit nötig sein würde: Die Kommunikationsschnittstelle zwischen Roboter und Prüfstand war nicht robust genug. Zuverlässigkeitsprobleme des Treibers vom CAN – Controller sowie eine allgemeine Instabilität der Dauerlaufsteuerung sorgten für zahlreiche Stillstände. Hinzu kamen Probleme durch Missverständnisse zwischen Prüfstandsentwickler und Bediener.

Ein Prüfstand für bewegliche Karosseriekomponenten muss sehr vielseitig sein. Für jeden Prüfling gelten in Abhängigkeit vom jeweiligen Fahrzeugtyp und Entwicklungsstand andere Voraussetzungen und Testbedingungen. Es ist eine große Motivation, ein Gesamtsystem zu entwickeln, welches unter diesen Bedingungen technisch zuverlässig und wirtschaftlich rentabel funktioniert. Nach dem Produkthaftungsgesetz muss ein Automobilhersteller Produkte von Zulieferfirmen ausnahmsweise nicht testen, wenn der Zulieferer das relevante Know – how zum Testen der Produkte nachweisen kann [Erb-06]. Das Ergebnis dieser Arbeit unterstreicht die hohe Kompetenz von Ingenieurdienstleistern, eigenständig umfangreiche Entwicklungsarbeiten auszuführen und damit einen unverzichtbaren Beitrag zu den hochgesteckten Wachstumszielen der Automobilkonzerne zu erbringen.

## 1.2 Zielsetzung

Das Versuchswesen für Karosseriekomponenten in der Fahrzeugentwicklung ist ein sehr spezialisiertes und vielseitiges Arbeitsumfeld. Trotzdem ist es im Vergleich zu anderen Entwicklungsbereichen weniger wissenschaftlich geprägt. Die Gründe hierfür sind, dass es neben technischem Grundlagenwissen besonders auf die Erfahrung der Mitarbeiter bei der Durchführung von Prototypentests sowie auf die korrekte Interpretation der Ergebnisse ankommt. Diese Arbeit leistet einen Beitrag dazu, die angewandten Versuchsmethoden ingenieurwissenschaftlich weiterzuentwickeln. Ein Ziel ist es, die Nachvollziehbarkeit und Reproduzierbarkeit zu verbessern und Versuchsergebnisse zukünftig stärker durch objektive Fakten begründen zu können.

Studenten erarbeiten im Rahmen von Praktika oder Studienabschlussarbeiten permanent neue Lösungsansätze für knifflige Problemstellungen, oftmals kann dabei die langfristige Verwendbarkeit der Konzepte aus Zeitgründen nicht mehr während der Bearbeitungsdauer nachgewiesen werden. Die für eine Praxistauglichkeit erforderlichen Optimierungen, die im Vergleich zum Gesamtaufwand in der Regel gering sind, bleiben aus. Um einen Lösungsansatz so umzusetzen, dass dieser langfristig und gewinnbringend eingesetzt werden kann, bedarf es mehrerer Entwicklungsschleifen und damit mehr Zeit. Ein weiteres Ziel dieser Arbeit ist es, aufgrund des mehrjährigen Bearbeitungszeitraums und der Nähe zur Praxis ein Prüfsystem zu entwickeln, das langfristig einsetzbar ist.

Die zunehmende Produkt- und Variantenvielfalt im Automobilbau erweitert die Anforderungen an die zur Validierung eingesetzten Prüfsysteme. Prüfstände mit der Beschränkung auf

einen speziellen Testfall, sogenannte Einzellösungen, sind wirtschaftlich nicht mehr vertretbar [Nat-10b]. Ein weiteres Defizit bei Zuverlässigkeitsversuchen ist die isolierte Prüfung einzelner Bauteile oder Baugruppen. Die Aussagekraft mit Hinblick auf das Gesamtsystem ist dann nur bedingt gegeben. Vor der Umsetzung der Inhalte dieser Dissertation waren die verwendeten Prüfstände im Karosserieversuch überwiegend auf die sogenannten *Topfunktionen* fokussiert [Ber-09]. Bei einem Fensterheberdauerlauf wurden ausschließlich die Topfunktionen *Fenster öffnen* und *Fenster schließen* im Prüfablauf berücksichtigt. Bedeutende *Subfunktionen*, wie *Fenster automatisch schließen* oder eine Überprüfung des Einklemmschutzes, fanden keine Relevanz. Das technische Ziel der Entwicklungsarbeit ist daher ein universelles und in der industriellen Praxis einsetzbares Testsystem für die Automatisierung von Komponentenversuchen. Basierend auf einer modularen Systemarchitektur soll flexibel auf sich verändernde Testbedingungen reagiert werden können. Das neue Konzept muss durch die Einbindung von Subfunktionen die Prüftiefe steigern. Eine Erhöhung des Automatisierungsgrads der Prüfung sowie eine Minimierung von Prüfungsunterbrechungen soll die Produktivität des Versuchswesens steigern. Spätere Erweiterungen und Optimierungen am Prüfsystem müssen mit vertretbarem Aufwand möglich sein, um mit den Innovationen in der Automobilentwicklung Schritt halten zu können.

Zur Bedienung des Prüfstands darf kein Ingenieurstudium erforderlich sein, vielmehr soll eine kurze Einarbeitung die intuitive Anwendung ermöglichen. Dies betrifft besonders die eingesetzte Robotik zur mechanischen Betätigung der Prüflinge. Roboter sind im Versuchswesen trotz der auf den ersten Blick vielfältigen Einsatzmöglichkeiten noch nicht sehr verbreitet. Diese Arbeit leistet einen Beitrag dazu, die Verwendung von Robotern im Versuchswesen zu etablieren. Das vorliegende Schrifttum vermittelt Ingenieuren im Versuchswesen ein kompaktes und verständliches Basiswissen zur Durchführung von Lebensdauerprüfungen am Beispiel von beweglichen Karosseriekomponenten. Nur wenn die Bediener die grundsätzliche Struktur erkennen und die Arbeitsweise des Prüfsystems nachvollziehen können, ist das gesamte Potential in der Praxis abrufbar.

Moderne mechatronische Produkte beziehen ihre hohe Funktionalität aus dem Zusammenspiel von Mechanik, Elektronik und im besonderen Maß der Informatik [Fri-07]. Umso erstaunlicher ist es, dass softwarebasierte Prüfstände zur Dauererprobung von Karosseriekomponenten wenig verbreitet sind. Mit dem Ziel, eine Weiterverbreitung zu forcieren, bietet diese Arbeit Softwareentwicklern ein Prüfsystemkonzept, welches auch abseits der Automobilindustrie verwendbar ist.

## 1.3 Vorgehensweise

Das in dieser Ausarbeitung dokumentierte Prüfsystemkonzept entstand im Verlauf einer fünfjährigen Entwicklungsphase. Die während meiner Diplomarbeit [Vog-07a] entwickelte Prüfstandssoftware wurde zunächst mit jeder weiteren Prüfung um neue Komponenten ergänzt. Mit Hilfe dieser Individuallösungen für konkrete Problemstellungen konnten grundlegende Ansätze für ein flexibles Automatisierungssystem konzipiert und validiert werden.

Nach [Pah-07] durchläuft die Entwicklung eines technischen Produkts bis zum Vorliegen des Endergebnisses mehrere Iterationsschleifen. Idealerweise ist die Schleifenanzahl minimal und die darin auszuführenden Änderungen sind gering. Der Gesamtaufwand einer Produktentstehung ist maßgeblich vom Wissen, Können und der Erfahrung des Entwicklers in seinem Arbeitsumfeld abhängig. Aus diesem Grund erfolgte parallel zur Erarbeitung des Prüfsystem-



konzepts eine intensive Mitarbeit in der Versuchsabteilung. Das Aufgabenfeld beinhaltete das Aufbauen, Inbetriebnehmen, Betreuen und Bewerten verschiedenster Prüfungen. Eine derartige Vorgehensweise ist sehr wichtig, da sich nur so ein Verständnis für die tatsächlich sinnvollen Funktionen eines Prüfstandssystems entwickeln kann. Die in dieser Arbeit dokumentierten mechanischen Bauteile durchliefen ebenfalls mehrere Entwicklungsstufen mit jeweils ausgedehnten praktischen Tests.

Als Herzstück dieses Automatisierungssystems wurde eine universelle und modular aufgebaute Softwarearchitektur konzipiert, entwickelt, validiert und erfolgreich in der Praxis etabliert. Die darin integrierte aufgabenorientierte Schnittstelle zur Einbindung von Robotertechnik, die Fähigkeit zur umfangreichen Analyse von Funktionsstörungen, die zweckmäßige Verwaltung und Präsentation der aufgenommenen Messdaten sowie die Informationsdienste tragen wesentlich zum betriebswirtschaftlichen Erfolg bei. Durch eine Schnittstelle zur Software *CANoe* der Firma *Vector Informatik* ist es gelungen, herstellerunabhängig einen Zugriff auf die Fahrzeugelektronik zu gewährleisten. Seit dem Praxiseinsatz dieser neuen Prüfsystemsoftware mit dem Namen *DaLaBert* in den Versuchswerkstätten waren nur geringe Anpassungen des Konzepts oder der Architektur nötig. Die Ergänzungen beschränkten sich hauptsächlich auf Prüfmodule und Softwaretools, die durch neue Prüfungen oder veränderte Anforderungen erforderlich wurden.

Das vorliegende Konzept ist nicht auf Basis einer schon zu Beginn eindeutig abgegrenzten Zielsetzung entstanden. Die Anforderungen weiteten sich während der Bearbeitungsdauer stetig durch neue Erfordernisse im Zusammenhang mit der jeweiligen Prüfung aus. Zwischenstufen in der Entwicklung erfuhren eine Validierung und Optimierung anhand realer Dauerversuche.

## 2 Reale und virtuelle Erprobung in der Fahrzeugentwicklung

Im Vorfeld der aufwendigen Entwicklung eines Prüfsystems für Realversuche an Fahrzeugkomponenten ist die Frage zu diskutieren, inwieweit diese Entwicklung zeitgemäß und für die Zukunft essentiell ist. Gibt es für derartige Systeme mittelfristig eine Verwendung, oder wird die reale Erprobung in naher Zukunft komplett durch die Erprobung auf Basis von Computermodellen ersetzt? Begriffe wie *Simulation* oder *Virtual Reality* werden in diesem Zusammenhang oft genannt. Die Antwort auf diese Frage ist nicht einfach. Für Entscheidungsträger in der Automobilindustrie ist die virtuelle Erprobung auf den ersten Blick eine hervorragende Möglichkeit, reale Prototypen und damit viel Zeit und Geld im Entwicklungsprozess einzusparen und die Wettbewerbsfähigkeit zu erhalten. Die rasante Weiterentwicklung der digitalen Datenverarbeitung hat diese Möglichkeit in den letzten Jahrzehnten geschaffen.

Publikationen, die sich mit der Einführung neuartiger Simulationsmöglichkeiten auseinandersetzen, beschreiben überwiegend sehr gute Übereinstimmungen zwischen Ergebnissen simulierter Versuche und parallel am realen Bauteil gemessenen Werten [Lie-04], [Ber-08], [Dai-09]. Demgegenüber sind Vergleiche in der Fachliteratur aufgeführt, die schon bei relativ einfachen Problemstellungen unter Anwendung der linearen Schadensakkumulationshypothesen Unterschiede von 1:50 aufweisen [Hai-06], [Sie-06], [Rad-07], [San-08]. Auch drängt sich der Verdacht auf, dass negativ bewertete Vergleiche zwischen der Simulation und einem Realversuch seitens der Automobilhersteller nicht im großen Umfang veröffentlicht werden. Die Wahrheit liegt irgendwo in der Mitte.

### 2.1 Geschichte der Fahrzeugerprobung

Im Jahr 1886 erteilte das Kaiserliche Patentamt dem Konstrukteur Carl Benz das Patent für den *Benz Patent – Motorwagen Nr. 1*. Dieses Gefährt erlebte seine erste öffentliche Probefahrt am 2. November 1886 in Mannheim. Dabei glich es eher einem dreirädrigen Kutschwagen als einem Automobil heutiger Bauweise. Schon damals wurde im Vorfeld dieses Ereignisses eine Erprobung durchgeführt, aus Gründen der Geheimhaltung wurde diese zunächst auf dem Firmengelände vorgenommen. Die erste nächtliche Ausfahrt endete nach wenigen Minuten und einhundert Metern aus technischen Gründen. Später vergrößerte sich die Reichweite schrittweise [Ben-86], [Ben-11b].

Gegen Ende des 19. Jahrhunderts war der Fahrzeugbau noch nicht automatisiert. Jede Einzelanfertigung eines Automobils erfolgte mit hohem handwerklichen Können. Notwendige Änderungen der Einzelteile wurden direkt bei der Montage vorgenommen. Mit der Einführung des Fließbandes im Jahr 1913 in die Fertigung des *Ford T (Tin Lizzy)* änderten sich die Bedingungen. Für die Fließbandfertigung waren eine ausgereifte und erprobte Konstruktion und darauf aufbauend wiederholgenaue Verfahren zur Fertigung der Einzelteile in zulässigen Toleranzen notwendig. Im Jahr 1934 kam erstmals im *Citroen 11CV* eine selbsttragende Karosserie in Serienfertigung zum Einsatz. Weiterentwicklungen in der Blechumformung machten dies möglich und sorgten zudem für gewünschte Gewichtseinsparungen [Dom-97], [Bra-01], [Deu-06].

Als Zuverlässigkeitsnachweis für Fahrzeuge verschiedenster Hersteller und für Vergleiche mit der Konkurrenz wurde im Jahr 1923 erstmals das *24 – Stunden – Rennen von Le Mans* veranstaltet. Neben dem *24 – Stunden – Rennen auf dem Nürburgring* gehört es noch heute zu den anspruchsvollsten und prestigeträchtigsten Langstreckenrennen. Bereits von 1900 – 1905 gab es mit dem *Gordon – Bennett – Cup* einen internationalen Leistungsvergleich zwischen mehreren Automobilmarken in sogenannten *Stadt – zu – Stadt – Rennen*. Die Fahrzeit betrug in der Regel nicht mehr als sieben Stunden [Aut-11], [Dav-11].

Bis in die 1930iger Jahre dominierten experimentelle Versuche sowie Erfahrungswerte der Entwickler die Automobilkonstruktion. Für die Berechnung von einfachen Zusammenhängen aus dem Bereich der Balkentheorie standen lediglich Rechenschieber, numerische Funktions- tafeln und mechanische Hilfsmittel zur Verfügung. Der von *Konrad Zuse* im Jahr 1941 als Z3 vorgestellte erste funktionsfähige Computer bildete die Grundlage für heute angewandte Berechnungsaufgaben, bei denen große Gleichungssysteme mit Millionen von Unbekannten zu lösen sind [Deu-06], [Schü-11].

1965 lag der Schwerpunkt der Lebensdauererprobung im Fahrversuch. Hinzu kam die Möglichkeit, maximale Biegebeanspruchungen des Vorderrad – Achsschenkels an gefährdeten Stellen zu berechnen. Die Durchführung von Betriebsfestigkeitsversuchen basierte auf Wöhlerlinien. Die Bestimmung der Spannungsverteilung mittels Dehnmessstreifen steckte noch in den Anfängen, da die sehr voluminöse Sensorik nur wenige Messstellen zuließ. Crashtests sind seit Anfang der 1960iger Jahre einer der wichtigsten Bestandteile der Erprobung, um das lebensrettende Verhalten von Fahrzeugen bei Unfällen zu erkunden und zu optimieren. Die passive Sicherheit avancierte zum wichtigen Verkaufsargument [Köp-01], [Bra-03], [Hau-04].

Seit Beginn der 1970iger Jahre spielt die Ausgewogenheit zwischen Fahrzeugmasse, Kraftstoffverbrauch und Abgasemission sowie die Insassensicherheit eine ganz entscheidende Rolle. Durch die Erdölverknappung in den Jahren 1973, 1974 und 1979 waren bessere Wirkungsgrade der Fahrzeuge nötig. Aufgrund der steigenden Verkehrsdichte und der damit verbundenen Umweltbelastung folgten gesetzliche Bestimmungen zum Abgasausstoß. Eine ansteigende Zahl von Todesopfern im Straßenverkehr hatte konstruktive Neuerungen für mehr Sicherheit zur Folge. Diese Innovationen waren unter anderem möglich, weil die Finite – Elemente – Methode in die rechnerische Simulation Einzug hielt. Steifigkeiten, Eigenfrequenzen und Spannungen einer Karosserie unter Betriebslast ließen sich nun mit vertretbarem Aufwand berechnen [Deu-06].

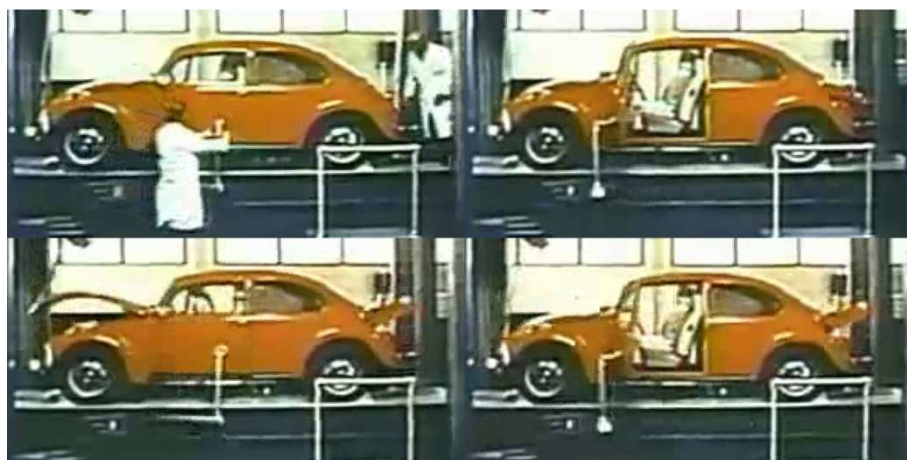


Abbildung 2: Ausschnitte eines Werbefilms zum VW Käfer 1303

Die hohe Bedeutung der Zuverlässigkeit eines Kraftfahrzeuges für den Kunden ist in einem Werbefilm des Jahres 1973 erkennbar. Unter dem Motto *Auf Herz und Nieren geprüft* wird

ein VW Käfer 1303 während der Dauererprobung auf einem Prüfstand abgebildet. Abbildung 2 zeigt vier Ausschnitte aus einem Video, welches die kombinierte mechanische Betätigung von Fahrertür, Front- und Heckklappe, Beifahrersitz sowie Fahrwerk demonstriert. Die relativ hohe Betätigungsfrequenz im Originalfilm lässt allerdings auf die Verwendung eines Zeitraffungsfaktors zu Gunsten der Werbewirksamkeit schließen.

Eine neue Ära in der Automobilentwicklung leitete die Erfindung des Mikroprozessors ein. Bis Mitte der 1970iger Jahre bestanden Steuerungseinheiten der Fahrzeugausstattung zum überwiegenden Teil aus mechanischen Bauteilen. Neue Funktionen waren Ergebnisse von immer komplizierter werdenden Hardwareentwicklungen. Durch die Mikroelektronik ergab sich die Möglichkeit, neuartige Funktionen durch eine Kombination aus Hard- und Softwarekomponenten zu integrieren. Auf Grundlage der ersten softwarebasierten Funktionen war es sehr einfach, weitere Funktionen zur Steigerung des Bedienkomforts hinzuzufügen. Ein Beispiel sind die *Maut – Funktionen* des elektrischen Fensterhebers. Bei einer etwas verlängerten Schalterbetätigung bewegt sich die Fensterscheibe eigenständig in ihre Endposition, ohne dass der Bediener den Schalter während des gesamten Bewegungsvorgangs halten muss. Für diese Zusatzeigenschaften sind lediglich Softwareerweiterungen nötig. Der Ansatz, Software anstelle von Hardware zu verwenden, bietet zudem großes Einsparpotential. Andererseits steigt durch das Verschmelzen von Hard- und Softwarekomponenten und die damit implizierten gegenseitigen Abhängigkeiten die Anzahl der möglichen Fehler. Mittlerweile ist es nicht mehr möglich, sämtliche Funktionen in jeder theoretischen Kombination manuell zu testen. Dies geschieht heute durch automatisierte systematische Tests, wodurch wiederum potentielle Fehlerquellen entstehen. Besonders in den Anfangsjahren der Elektronisierung hat diese Thematik massive Elektronikprobleme verursacht [Züh-04], [Rie-07].

Seit Mitte der 1990iger Jahre wird die HIL – Technologie angewendet. Hierbei besteht der zur Abstimmung oder Erprobung vorgesehene Teil des Prüfaufbaus aus realen Bauteilen, die übrigen zum Betrieb benötigten Komponenten werden mittels Software simuliert. Beim Test eines Motorsteuergeräts erfolgt beispielsweise die Beschaltung der Eingänge mit realitätsnahen simulierten Sensorsignalen, um die Reaktion des Steuergeräts an den Ausgängen zu analysieren. Dieses Verfahren bietet die Möglichkeit, gleichbleibende Fahrsituationen mit variablen Steuergerätparametern zu testen. Aufwendige Hardwareprüfstände sind dazu nicht nötig. Auch können Fahrsituationen in Grenzbereichen sehr realistisch erprobt werden, ohne dabei Fahrzeug oder Fahrer zu gefährden [Deu-06].

Fahrzeuge müssen bei extremen Klimabedingungen, weltweit unterschiedlichen Kraftstoffqualitäten und allen erdenklichen Nutzerprofilen über einen langen Zeitraum zuverlässig funktionieren. Qualitätsdefizite werden durch international vernetzte Medien und Kunden schnell bekannt und können bei einer Häufung einen großen Imageschaden für den Hersteller bedeuten. Um diesen zu verhindern, werden heute viele Millionen Testkilometer in extrem warmen, kalten, staubigen, trockenen und nassen Gebieten gefahren. Bei der im Jahr 2007 auf den Markt gebrachten *Mercedes Benz C – Klasse* kamen so 24 Millionen Testkilometer zusammen [Bri-07].

Einzelkomponenten werden heute Vibrationsprüfungen auf Shakeranlagen unterzogen. Die Schwingungsprofile dafür entstammen einer Aufzeichnung von der realen Teststrecke. Dem Sonnenlicht ausgesetzte Komponenten müssen in einer Sonnensimulation extremen Strahlungen standhalten. Die virtuelle und reale Erprobung findet in der mittleren Phase der Entwicklung parallel statt. Erst wenn die Ergebnisse der virtuellen Welt zufriedenstellend sind, erfolgt eine Verifizierung durch Realversuche. Besonders bei Crashversuchen hat dieses Vorgehen große Kosteneinsparungen ermöglicht. Allerdings ist die Erprobungsphase heute nicht mehr mit dem Produktionsstart abgeschlossen. Zeigt sich nach der Erstausslieferung ein Bedarf an

Änderungen, werden diese bei der ersten Inspektion durchgeführt, ohne dass es vom Kunden bemerkt wird. Trotz der heute verfügbaren Methoden der breit angelegten und besonders im Elektronikbereich hoch automatisierten Tests ist es nicht gänzlich auszuschließen, dass sämtliche für die Kundennutzung relevanten Prüffälle bekannt und erfasst sind. Ein kleines Restriktio bleibt bestehen. Erst die sogenannte *Erprobung durch den Endkunden* und Rückmeldungen aus den Werkstätten nach einer mehrjährigen Laufzeit des Fahrzeugs bringen letztendlich Gewissheit [Fah-10]. Abbildung 3 zeigt die Entwicklung von der werkstatorientierten Konstruktion hin zur virtuellen Produktentwicklung.

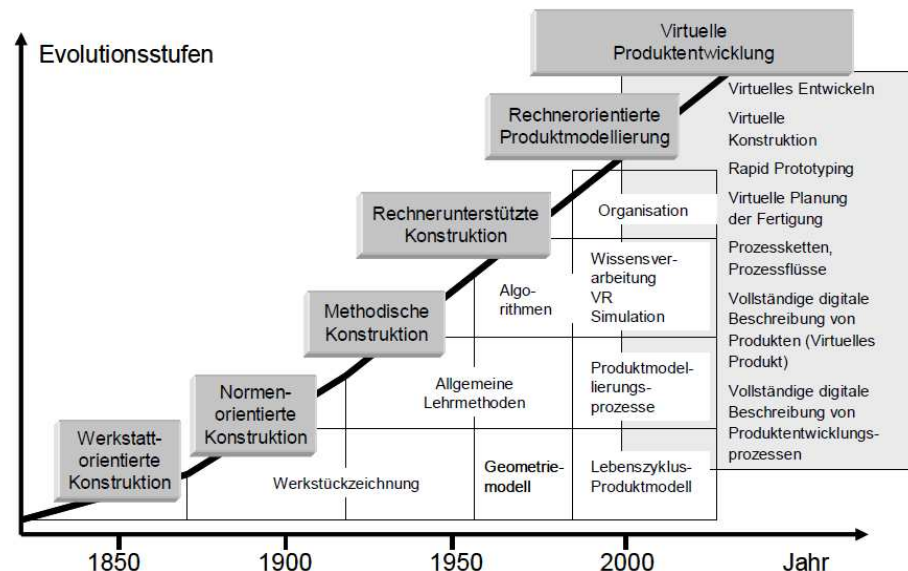


Abbildung 3: Evolutionsstufen der Konstruktion [Her-07]

Abbildung 4 beschreibt die zeitliche Positionierung der einzelnen Erprobungsschritte während des Entwicklungsprozesses. Dabei wird die Überschneidung von Simulation und realen Prüfstandsversuchen deutlich sowie die Fortsetzung der Erprobung über den Produktionsstart *SOP* hinaus.

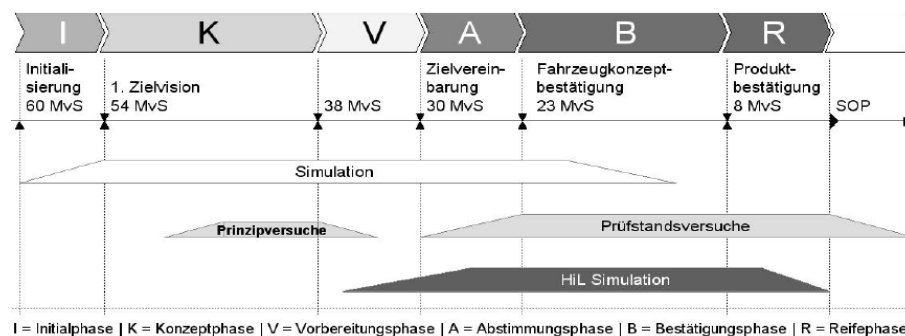


Abbildung 4: Zuordnung von Erprobungsschritten zu Entwicklungsphasen [Deu-06]

## 2.2 Virtuelle Entwicklung und Erprobung

Der Einsatz von virtuellen Modellen in der Fahrzeugentwicklung geht zurück in die 1960iger Jahre und lässt sich mit der damaligen Leistungssteigerung der Computer begründen. Im Jahr 1982 galt es noch als Zukunftsvision, basierend auf gemeinsam genutzten Daten eine Verknüpfung zwischen Konstruktion, Berechnung und Versuch herzustellen, um eine Steigerung von Effektivität und Effizienz im Produktentwicklungsprozess zu erreichen. Diese Vision ist

heute so wirklich wie unverzichtbar, da die Betriebsfestigkeitsuntersuchungen durch Aufwand und Dauer einen kritischen Pfad im Entwicklungsprozess darstellen. Nur mit Hilfe von digitalen Hilfsmitteln kann dieser heute im Zeit- und Kostenrahmen durchlaufen werden, ohne dabei Qualitätsverluste oder Sicherheitsprobleme zu riskieren. Abbildung 5 zeigt die Entwicklungstendenz und den Zusammenhang von Simulation, Versuch und Entwicklungszeit [Jun-04], [Bra-07], [Ata-08].

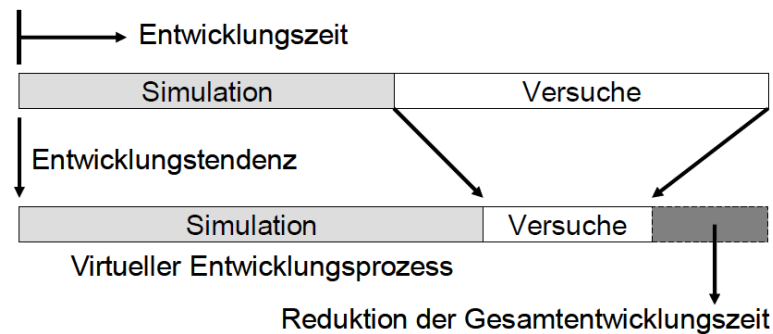


Abbildung 5: Tendenz zur Reduktion der Gesamtentwicklungszeit [Her-07]

Grundsatzentscheidungen fallen heute in der Konzeptphase einer Fahrzeugentwicklung nicht mehr auf der Basis von physikalischen Prototypen, sondern mit Hilfe von virtuellen Modellen. Dazu zählt die Festlegung auf eine der möglichen Varianten mit Eigenschaften wie Gewichtsverteilung, Einbauposition des Motors, Schwerpunktlage, Fahrdynamik und Crashverhalten. Während der darauf folgenden Designphase spielt die virtuelle Realität eine ebenso wichtige Rolle. Dort hilft diese bei der Entscheidungsfindung über innere und äußere Designformen. Die Prototypen- und Validationsphase ist gekennzeichnet durch eine Verfeinerung der virtuellen Modelle bis zu dem Detaillierungsgrad, welcher den exponentiell ansteigenden Zeit- und Kostenaufwand noch rechtfertigt. Der Reifegrad dieser Modelle ist heute dank der virtuellen Erprobung sehr hoch, die Anzahl der simulierbaren Effekte nimmt stetig zu. Liegen in der Prototypenphase die ersten realen Bauteile vor, sind diese mit den Anforderungen des Lastenhefts abzugleichen. Anhand der Testergebnisse lässt sich die Qualität der virtuellen Versuche beurteilen. Es ist jedoch zu berücksichtigen, dass Prototypen gegenüber Serienteilen in der Regel noch Defizite sowohl in der Materialqualität als auch beim Herstellungsprozess aufweisen. In dieser Phase ist im Besonderen auf die Vergleichbarkeit von Rechenmodellen der verschiedenen Berechnungsdisziplinen zu achten. Diese kann Mängel aufweisen, wenn die Konsistenz zwischen CAD- und FEM – Daten nicht mehr durchgängig gewährleistet ist. Je komplexer das Produkt, desto anfälliger ist es für Änderungen in späten Phasen der Entwicklung. Der höchste Reifegrad der virtuellen Modelle wird erst während der Produktionsphase erreicht [Bre-06], [Her-07], [Dai-09].

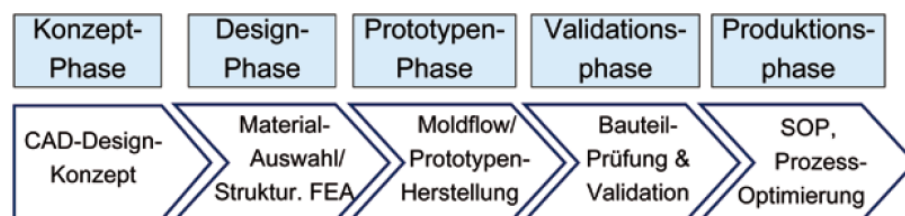


Abbildung 6: Phasen im Entwicklungsprozess eines Fahrzeugs [Gla-09]

Die Vorteile der Simulation im Vergleich zum physikalischen Modell liegen darin, Potenziale verschiedener Varianten sehr viel schneller zu erkennen. Zudem lassen sich Details im Modell mit geringem Aufwand variieren oder austauschen. Änderungen auf Basis der ersten Vorschläge können durch virtuelle Werkzeuge in einem Bruchteil der Zeit angepasst werden, die ein reales Bauteil dafür in Anspruch genommen hätte. Zudem besteht keine Bindung an physika-

lische oder technische Zwänge. Die Berechnung von Spannungen ist beispielsweise an jeder beliebigen Stelle im Bauteil möglich, was in der Realität besonders an den kritischen Stellen nur äußerst selten der Fall ist. Auf Basis der Berechnungsergebnisse ist heute die automatisierte Optimierung von Modellen möglich. Bei den Untersuchungen zur Dauerfestigkeit mit Mitteln der Simulation steht der Verweis auf kritische Stellen im Vordergrund, nicht die Vorhersage einer Lebensdauer [Deu-06].

Die rechnergestützte Entwicklung *CAE* umfasst eine Vielzahl von Softwarewerkzeugen. Im Fahrzeugbau besonders bewährt sind die Werkzeuge zur Konstruktion *CAD*, für digitale Versuche *DMU*, zur Dynamik- und Kinematiksimulation *MKS* sowie zur Anwendung von Finite Elemente Methoden *FEM*. Die Basis für diese weiterführenden digitalen Werkzeuge ist das rechnerbasiert konstruierte Fahrzeugmodell. Dieses wird mittels *CAD* – Software in seinen vielen Einzelteilen erstellt und anschließend zu einem komplett virtuellen Fahrzeug zusammengefügt. Durch Softwaretools, die auf das *DMU* spezialisiert sind, können anschließend Ein-, Ausbauuntersuchungen, Kollisionsprüfungen und Baubarkeitsprüfungen durchgeführt werden. Die Mehrkörpersimulation *MKS* liefert Erkenntnisse über das Verhalten von dynamisch beanspruchten Komponenten bei wirkenden Kräften, Geschwindigkeiten und Beschleunigungen. Mittels der *FEM* – Methoden kann eine Aussage getroffen werden, welche Spannungen und Verformungen durch die auftretenden Kräfte an der Konstruktion zu erwarten sind [Mey-07b], [Sei-08].

Eine Weiterentwicklung der genannten Werkzeuge vollzieht sich in drei verschiedenen Bereichen. Zum einen betrifft diese die Verbesserung und Rechengeschwindigkeit der Software. Des Weiteren besteht das Bestreben, das gesamte System als Modell abzubilden und nicht nur einzelne Komponenten. Der dritte Entwicklungsbereich zielt auf die Optimierung der Modelle und Simulationsmethoden, ein Beispiel dafür ist die stochastische Simulation [Jun-04]. Dieser Ansatz berücksichtigt, dass in der Realität die Streuung von Systemeigenschaften, beispielsweise Blechdicke, Materialkennwerte oder Geometrien, einen nicht zu vernachlässigenden Einfluss auf die Bauteileigenschaften hat. Allerdings sind durch die benötigte Rechenleistung Grenzen gesetzt [Str-02]. Ebenso sind direkte Vergleiche zwischen der Simulation und realen Versuchen eine Möglichkeit zur Verbesserung der Modelle. In einer Variante wird ein realer Crashtest mit Methoden der Fotogrammetrie und Hochgeschwindigkeitskameras aufgezeichnet. Im Anschluss erfolgt eine Überlagerung des aufgenommenen Films mit der Aufzeichnung eines virtuellen Crashtests. Abweichungen zwischen beiden Testverfahren werden ohne aufwendige Messungen auf den ersten Blick sichtbar. Diese Methode beschränkt sich aber auf die Vergleichbarkeit oberflächlich sichtbarer Elemente [Rag-05].

Beim Homologationsprozess ist die virtuelle Erprobung mittlerweile ein fester Bestandteil. Sie ist jedoch abhängig von länderspezifischen Anforderungen und einer Abstimmung mit den Behörden. Mit dem Ziel, die Wettbewerbsfähigkeit der europäischen Automobilhersteller zu stärken, hat die Gruppe *CARS21* im Jahr 2005 einen Plan vorgestellt, mit dessen Hilfe bis ins Jahr 2015 die Ergebnisse aus virtuellen Testverfahren beim Freigabeprozess anerkannt werden sollen. Dadurch könnte der Freigabeprozess schon vor der Prototypenphase beginnen und zur Zeit- und Kostenersparnis beitragen. Als Voraussetzung ist eine Überprüfung der Berechnungsmethoden durch Realversuche nötig. Der *TÜV Süd* sieht diesbezüglich vor, parallel zur Simulation Laborversuche durchzuführen oder Serienteile nachzuprüfen. In jedem Fall wird beim Entwickler ein erhebliches Maß an Fachkompetenz und Erfahrung vorausgesetzt. Zudem müssen für reale und virtuelle Versuchstypen die gleichen Anforderungen gelten. Nutzfahrzeughersteller profitieren besonders davon, da in dieser Branche wesentlich geringere Stückzahlen produziert werden. Bereits seit über zehn Jahren ist es seitens der Prüfbehörden anerkannt, den Nachweis gegen das Umstürzen von Reisebussen durch virtuelle Werkzeuge zu erbringen [Gus-06], [Lec-06].



In der Praxis hat die Ausweitung der virtuellen Entwicklung eine große Datenmenge zur Folge. Bei der *Mercedes Benz – C – Klasse* aus dem Jahr 2006 betrug das Datenvolumen der kompletten Fahrzeugmodellierung 2.130 Gigabyte. Die verstärkte Aufteilung der Entwicklungsumfänge auf Ingenieurdienstleister und Zulieferfirmen führt zu einer dezentralen Entstehung der Modelldaten. Zudem besitzen die Fahrzeugmodelle Varianten und Derivate, die in vielen Iterationsschritten rechnerisch zu bewerten sind. Um unter diesen Voraussetzungen den Überblick zu wahren und Dateninkonsistenzen zu vermeiden, sind mit den virtuellen Werkzeugen verknüpfte Datenmanagementsysteme unverzichtbar [Bri-07], [Naw-08].

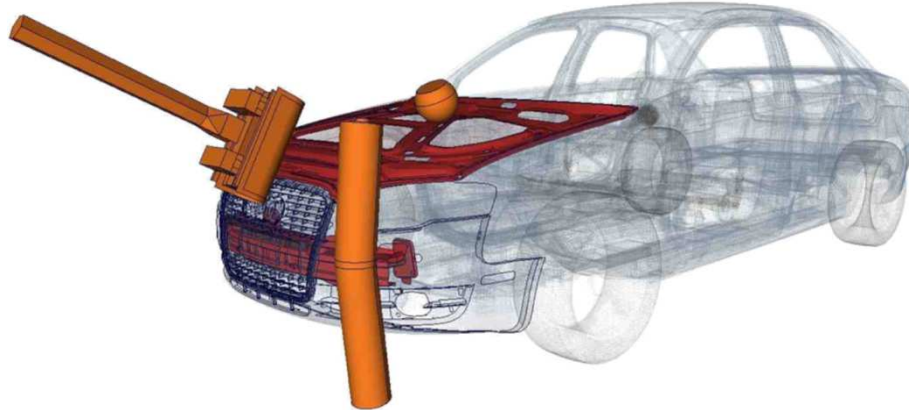


Abbildung 7: Simulation verschiedener Crashesituationen

## 2.3 Grenzen der Simulation

Generell sind der rechnergestützten Simulation Grenzen gesetzt in Form von limitierter Rechnerkapazität, Zeit und finanziellen Mitteln. Ein Modell sollte im Aufbau möglichst einfach sein, es hat jedoch viele Anforderungen zu erfüllen, um aussagekräftige Ergebnisse liefern zu können. Jedes Modell ist grundsätzlich im Zusammenhang mit seiner Entstehung zu betrachten. Ändern sich die Voraussetzungen, ist das Ergebnis möglicherweise nicht nur fehlerhaft, sondern gänzlich unbrauchbar [Buc-11].

Ein Hauptproblem der realen Erprobung im Entwicklungsprozess ist die Verfügbarkeit von seriennahen prüffähigen Bauteilen. Dies lässt sich gleichermaßen auf die Simulation übertragen. Auch hier steigt die Qualität und Detaillierung der Modelle erst mit der Zeit. In frühen Modellphasen bestehen daher Unsicherheiten aufgrund unvollständiger Datenstände. Eine häufige Folge ist, dass die Berechnungsergebnisse nicht mehr mit dem zwischenzeitlich verbesserten Entwicklungsstand des Produkts übereinstimmen [Mer-06], [Ber-08].

Technisch betrachtet sind der Simulation zwei unterschiedliche Arten von Grenzen gesetzt. Die erste Grenze ergibt sich aus der Technologie. Bei der Beurteilung virtueller Fahrzeuge werden der Tastsinn, Hörsinn und Geruchssinn nicht stimuliert, es kommt zu einer verfälschten Wahrnehmung des Gesamteindrucks und möglicherweise zu falschen Entscheidungen. Besonders im Premiumsegment erwarten Kunden ein hochwertiges haptisches Gefühl, einen satten Klang, kein störendes Knarzen oder Klappern sowie ein angenehmes Geruchsbild. Knarz- und Klappergeräusche verursachen beim Neuwagenhändler hohe Gewährleistungskosten, die Ursache ist oft nur mit großem Zeitaufwand zu finden. Die Simulation von hochfrequenten Geräuschen ist ebenfalls sehr aufwendig. Eine weitere technologiebedingte Beschränkung ist die ohnehin sehr schwierige Objektivierung der Ergebnisse. Beim Komfort eines Sitzes kann durch Berechnungen allenfalls die Festigkeit des Schaums vorhergesagt wer-



den. Für das subjektive Komfortempfinden bedarf es der Meinung einer Reihe anatomisch unterschiedlicher Probanden [Neu-04], [Bra-07]. Die zweite Grenze entsteht durch eine unzureichende Evaluation, zu geringe Detaillierung, Instabilität der Modelle, fehlendes physikalisches Verständnis oder eine schlechte Qualität der Eingangsparameter. Qualitätsmängel von Modellen hängen oftmals mit der Idealisierung von Material, Belastung und Geometrie sowie der damit verbundenen Vernachlässigung von Nichtlinearitäten zusammen [Jun-04], [Pie-09].

Ebenso problematisch ist der rechenzeitbedingte Verzicht auf Einflüsse durch Fertigungstoleranzen. Beispiele sind Abweichungen von der Nominaldicke durch Fließen beim Tiefziehen, Rückspringverhalten sowie Eigenspannungen nach dem Tiefziehen oder Verzug durch thermische Einflüsse beim Schweißen [Jun-04], [Voß-08]. Besonders zu beachten sind streuende oder unbekannte Materialparameter. Metalle sind heute gut simulierbar, da alle wesentlichen Eigenschaften erforscht sind. Die Querkontraktionszahl ist in allen Richtungen gleich. Bei Verbundwerkstoffen, wie dem kohlenstofffaserverstärktem Kunststoff *CFK*, gilt das nicht. Der Einsatz von derartigen neuen Materialien stellt eine große Herausforderung für die Simulation dar. Neue Werkstoffe benötigen viel Zeit und Erfahrung, bis die Parametereinstellungen annähernd angepasst sind. Ein besonderes Augenmerk gilt bei Kunststoffen einer Betrachtung des Materialverhaltens bei extremen Temperaturbereichen von -30 bis +80 Grad Celsius und unterschiedlichen Luftfeuchten. Hohe Dynamik ist ebenso ein großer Unsicherheitsfaktor in den Modellen [War-04]. Neben der Betrachtung eines einzelnen Materials stellt das Verhalten zweier Materialien zueinander eine Herausforderung dar. Beispiele in der Akustiksimulation sind das Bewegen der Fensterscheibe auf der Dichtung oder das Verhalten der Dichtung beim Türzuschlag. Eine weitere Unzulänglichkeit der Simulation ist die Betrachtung von Verschleiß- und Alterungsverhalten. In der Praxis wird die Betriebsfestigkeit einer Tür durch einen einmaligen Türzuschlag mit erhöhter Schließgeschwindigkeit untersucht. Die Einbeziehung von Lebensdauerkurven auf Basis des Wöhlerversuchs findet aus Zeit- und Kostengründen nicht statt.

Für zusätzliche Unsicherheit ist die große Anzahl von angewendeten Simulationsprogrammen und Modellen verantwortlich, folglich sind die Ergebnisse in vielen Fällen nicht vergleichbar. Oftmals sind die Unterschiede zwischen den Ergebnissen zweier verschiedener Programme bei der Simulation des gleichen Modells sehr groß. Die Qualität und Aussagekraft der Berechnungsergebnisse hängt im Wesentlichen vom Wissen und der Erfahrung des Bedieners ab [Gös-01]. Untersuchungen zeigen, dass bei einer nur gering veränderten Mittelspannung als Eingangsgröße für ein Modell die reale Abweichung um den Faktor 0,25 – 7,0 gegenüber dem berechneten Wert schwankt [Rad-07]. Wie das Beispiel des *Airbus A380* zeigt, kann bei der Verwendung unterschiedlicher Konstruktionsprogramme in einem Projekt auch eine fehlerhafte Datenkonvertierung zum Verhängnis werden. Das Fehlen von wenigen Zentimetern Kabel beim Zusammenfügen von Rumpfteilen verursachte hier aufgrund von Neuverkabelungen und Lieferverzögerungen Kosten in Milliardenhöhe.

Neben den technologischen und technischen Grenzen der Simulation gibt es mit der Produkthaftung eine gesetzliche Beschränkung. Bei der Produkthaftung haftet der Hersteller für Sach- oder Personenschäden, die auf ein fehlerhaftes Produkt zurückzuführen sind. Der Hersteller muss nicht haften, wenn nach dem objektiven Stand der Wissenschaft und Technik niemand den Fehler hätte verhindern können. Würde nun, trotz der in diesem Kapitel aufgeführten Defizite und Grenzen der Simulation, die reale Erprobung ganz oder in Teilen als freigaberelevante Absicherung entfallen, würde eine mögliche Instanz zur Fehlererkennung übergangen. Der Hersteller müsste im Fall von Schäden haften. Dies bedeutet neben strafrechtlichen Konsequenzen auch einen enorm großen Imageschaden [Erb-06].

## 2.4 Fortschritt durch Symbiose beider Verfahren

Wird die Simulation den realen Versuch irgendwann komplett ersetzen? Diese eingangs gestellte Frage lässt sich in dieser Arbeit nicht abschließend beantworten. Der Ausspruch von *Carl Benz: Das Automobil ist fertig entwickelt. Was kann noch kommen?*, stammt aus dem Jahr 1920 [Abe-02]. Seitdem hat sich sehr viel verändert und in den kommenden Jahrzehnten wird die Anzahl der Innovationen voraussichtlich steil ansteigen. Die Vergangenheit hat oft gezeigt, dass nicht für möglich gehaltene technische Vorstellungen irgendwann Realität werden. Diese positive Entwicklung wird auch die Erprobungsmethoden technischer Systeme betreffen.

Vieles spricht dennoch dafür, dass Realversuche langfristig einen festen Platz in der Fahrzeugentwicklung behalten werden. Davon sind die Verfasser der Quellen [Köp-01], [Pas-02], [Jun-04], [Deu-06], [Gra-06], [Her-07], [Boc-08], [Sei-08] und [Gla-09] überzeugt. Mit Hilfe der Simulation ist es heute möglich, Teilkomponenten eines Systems durch virtuelle Modelle mit einem bereits sehr hohen und stetig steigenden Reifegrad zu validieren. Aussagen über das Zusammenspiel der Teilkomponenten im Gesamtsystem und die hervorgerufenen Wechselwirkungen lassen sich nicht mit der benötigten Zuverlässigkeit treffen. Hierfür ist die reale Erprobung unabdingbar notwendig. Diverse praktische Erfahrungen von Mitarbeitern im Versuchswesen haben übereinstimmend ergeben, dass zwischen Modellen und Prototypen für die ersten Realversuche teilweise große Abweichungen bestehen.

Für eine Zulassung im Straßenverkehr sind reale Versuche zur Absicherung der Qualität und Sicherheit international vereinbart und gesetzlich vorgeschrieben [Kra-09]. Während der Erstellung dieser Arbeit wurden zahlreiche Rückrufaktionen im Automobilbereich öffentlich, oft im Zusammenhang mit verletzten und getöteten Menschen. Laut Kraftfahrt – Bundesamt gab es in Deutschland im Zeitraum von 2008 bis 2010 insgesamt 473 Rückrufe [Kba-10]. Für die Automobilhersteller sind diese Vorfälle neben dem enormen Imageschaden besonders in den USA oftmals mit hohen Schadensersatzforderungen verbunden. Eine umfangreiche und technologisch zeitgemäße Erprobung ist unter Berücksichtigung dieser Aspekte unverzichtbar.

Ziel muss es sein, die Vorteile der virtuellen und realen Erprobung zu verbinden, sodass beide Verfahren voneinander profitieren. Nur durch die Kombination von Berechnung und Experiment ist eine Steigerung der Effizienz des Entwicklungsprozesses im Gesamten möglich.



Abbildung 8: Verbindung von realer und virtueller Erprobung [Boc-08]

Beide Disziplinen haben im Bereich der Absicherung individuelle Stärken und Schwächen. Mit Hilfe frühzeitig verfügbarer Simulationsergebnisse kann rechtzeitig auf kritische Stellen hingewiesen werden, um einen hohen Reifegrad der ersten Prototypen zu erhalten und erhebliche Zusatzkosten zu verhindern. Nur der Realversuch prüft die nicht simulierbaren Effekte sowie die Funktionalität des Gesamtsystems und liefert dadurch verbesserte Eingangsgrößen und Daten zur Validierung virtueller Modelle. Eine Simulation ist niemals besser als ihre Eingangswerte.

Diese Arbeit leistet einen wesentlichen und notwendigen Beitrag zur Weiterentwicklung der realen Versuchstechnik. Eine einseitige Entwicklung würde aufgrund der gegenseitigen Abhängigkeit beider Versuchstechniken nicht zu einer allgemeinen Verbesserung der Erprobungsmethoden führen.

### 3 Stand der Technik und Forschung

Das in dieser Arbeit beschriebene mechatronische Prüfsystemkonzept tangiert eine Vielzahl von technischen Gebieten. Dazu zählen Prüfstandsautomatisierung, Lebensdauererprobung mit den zugrunde liegenden Vorschriften, Messdatenerfassung und Analyse, Robotik und Roboterwerkzeuge, Fahrzeugelektronik sowie Softwareentwicklung und -architektur.

#### 3.1 Prüfstandssysteme in der Automobilindustrie

Die Entwicklung von Prüfsystemkonzepten ist ein eher kleines Gebiet innerhalb der universitären Forschung. Der Grund mag darin liegen, dass Prüfstandssysteme weniger durch Methoden der Grundlagenforschung, sondern mehr auf Basis langjähriger Erfahrungen in der industriellen Praxis realisiert werden. Als Anbieter für ein breites Spektrum von Hard- und Softwarekomponenten sowie als Veranstalter von Kongressen für praxisnahe Prüfstandstechnik ist besonders die Firma *National Instruments (NI)* zu nennen. Darüber hinaus liegen leider nur wenige Publikationen zu diesem Themengebiet vor. Detaillierte technische Hintergründe gehören zum Firmengeheimnis und werden, wenn überhaupt, nur ansatzweise beschrieben. In den Quellen [Nat-06], [Fri-07], [Nat-09a] und [Nat-10a] sind Publikationen aufgeführt.

Zur Verdeutlichung des Ziels dieser Arbeit ist eine Unterteilung der eingesetzten Prüfstandssysteme in Standardprüfstände und Einzellösungen sinnvoll.

##### 3.1.1 Standardprüfstand

Als Standardprüfstand bezeichnete Systeme werden entweder in großer Anzahl verwendet oder durch modularisierte Standardbaugruppen zusammengesetzt. Beispiele sind Motor-, Getriebe- und Rollenprüfstände, Windkanal, Hydropulsanlage, Shaker, Klima- oder Regenkammer sowie Crashtestanlagen. Die Auslegung der Prüfstände erfolgt für einen langen Betriebszeitraum, Anpassungen an nachfolgende Produktreihen sind möglich. Verwendung finden diese Systeme in der Entwicklung und Produktion.

Die Entwicklung und der Vertrieb dieser Anlagen erfolgt durch internationale Firmen, welche in der Regel als Zulieferer für alle Automobilkonzerne fungieren. Die österreichische Firma *Anstalt für Verbrennungskraftmaschinen List (AVL)* mit circa 4.500 Mitarbeitern und einem Umsatz von 650 Millionen Euro im Jahr 2010 ist das weltweit größte Unternehmen dieser Art [Avl-10]. An der Entwicklung der Prüfstände ist eine große Anzahl an Mitarbeitern beteiligt, in Absprache mit dem Kunden werden speziell abgestimmte Komponenten verbaut oder individuelle Einstellungen vorgenommen. Das Bedienpersonal erhält umfangreiche Schulungen.

Die Automatisierung der Anlagen basiert nicht mehr ausschließlich auf speicherprogrammierbaren Steuerungen. Aus Gründen der Flexibilität, Verfügbarkeit und Wirtschaftlichkeit übernehmen PC – basierte Steuerungen verstärkt die Automatisierung [Bur-02]. Standardsoftwarepakete für die Automatisierung von Prüfsystemen sind *TestStand (NI)*, *Morphee (D2T)*, *TestCASE (Softing)*, *TwinCAT (Beckhoff)* oder *ProfiSignal – Klicks (Delphin)*. Diese Programme verfügen in der Regel über eine Auswahl von vordefinierten Prüfmodulen für Komponenten-

tests und Schnittstellen zu gängigen Hard- und Softwarekomponenten. Die größte Herausforderung bei der Entwicklung derartiger Softwarelösungen ist die richtige Balance zwischen einer geringen Komplexität in der Bedienung und hoher Flexibilität in der praktischen Anwendung.



**Abbildung 9: Standardprüfstände**

Abbildung 9 zeigt eine befahrbare Klimakammer, einen Motorenprüfstand, einen Hinterwagenabschnitt während der Vibrationsprüfung sowie ein Fahrzeug auf dem Rollenprüfstand im Produktionswerk.

Die Firma *Bertrandt* hat vor einigen Jahren einen Prüfstand für Türen auf Basis eines der genannten Softwaretools aufgebaut. Beim praktischen Einsatz zeigte sich, dass für jeden Dauerlauf mehr oder weniger tiefgreifende Änderungen zur Adaption eines neuen Fahrzeugtyps nötig waren. Schon nach wenigen Prüfungen verlor das System seine Übersichtlichkeit durch Modifikationen und Hilfskonstruktionen. Der Praxiseinsatz war nicht rentabel. Weitere Anbieter zeigten kein Interesse an einer Kooperation, da die Ansprüche an ein derartiges Prüfsystem in Bezug auf Funktionsumfang und Flexibilität zu zeit- und kostenaufwendig im Hinblick auf die zu erwarteten Verkaufszahlen gewesen wären.

### 3.1.2 Einzellösung

Einzellösungen sind in diesem Zusammenhang Prüfstände, die nur bedingt universell einsetzbar sind. In der Regel basiert die Entwicklung auf einer konkreten Anforderung, für die noch keine Prüfmöglichkeit vorhanden ist. Weitere Gründe können fehlende oder in Entstehung befindliche Prüfvorschriften, spezielle oder akute Probleme sowie die Absicherung neuartiger Funktionen sein. Der Zeitraum bis zur erwarteten Abgabe des Ergebnisberichts ist im Allgemeinen relativ kurz, die zur Verfügung stehende Zeit und die finanziellen Mittel sind sehr knapp bemessen. Als Basis für Einzellösungen dienen überwiegend die Softwaretools *LabVIEW (NI)* oder *MATLAB (The MathWorks)* sowie speicherprogrammierbare Steuerungen.





**Abbildung 10: Einzellösungen**

Abbildung 10 zeigt Prüfstände für Außenspiegeltest, Steifigkeitsmessungen an Fahrzeugtüren, Staubuntersuchungen und Türschlosserproben.

Oftmals vollzieht sich die Entwicklung dieser Prüfstände in enger Zusammenarbeit mit dem späteren Bedienpersonal. Zeitintensive Bestandteile der Programmierung, beispielsweise Mechanismen zur Verhinderung von Fehlereingaben, müssen so nicht im maximalen Umfang implementiert werden. Nur unter diesen Bedingungen sind Prüfungen mit einem Auftragswert von wenigen Tausend Euro wirtschaftlich rentabel umzusetzen. Einzellösungen finden sich fast ausschließlich in der Entwicklung und nicht in der späteren Produktion.

Das in dieser Arbeit vorgestellte Prüfsystemkonzept bildet für Anwendungen in der Komponentenerprobung eine Brücke zwischen beiden Arten von Prüfstandssystemen. Die Neuentwicklung eines Standardprüfstands ist aufgrund der am Markt verfügbaren Lösungen und der begrenzten Zeit im Rahmen einer Promotion nicht durchführbar. Das Ziel ist eine hard- und softwaretechnische Basis, mit deren Hilfe die aktuell verwendeten Einzellösungen in einem Konzept kombiniert werden können. Die passgenaue Abwägung zwischen der Flexibilität in der Anwendung und dem Abstraktionsgrad der Bedienoberfläche ist dabei die größte Herausforderung.

## **3.2 Betriebsfestigkeitsuntersuchungen von Karosserieteilen**

Die entwicklungsbegleitende Erprobung hat sich aufgrund der verkürzten Entwicklungsdauer eines Fahrzeugs in den letzten Jahren tendenziell zu einer Erprobung parallel zur Vorserienproduktion gewandelt. Zukünftig ist zu erwarten, dass die ersten realen Versuche unmittelbar vor dem Produktionsstart durchgeführt werden. Die Fertigung der Prüfteile erfolgt dann bereits mit Serienwerkzeugen. Bedingt durch die höhere Ausgereiftheit der Werkzeuge steigen die Qualität und Aussagekraft der Testergebnisse. Andererseits lassen sich detektierte Mängel nicht mehr rechtzeitig vor dem Produktionsstart abstellen. Die Herausforderung, innerhalb der heutigen Entwicklungszyklen und unter Berücksichtigung des stetig steigenden Funktionsumfangs ein vollständig ausgereiftes und fehlerfreies Fahrzeug zu entwickeln, ist immens. In

Ausnahmefällen ist es daher notwendig, nach dem Produktionsstart erkannte Mängel beim ersten Kundendienst zu beheben. Wirken sich diese Mängel schon vorher negativ aus, führt dies zu einer Unzufriedenheit beim Kunden und bei einer Häufung zur Beschädigung des Images der Marke. Erweisen sich Mängel aus der späten Absicherung als äußerst gravierend, drohen im schlimmsten Fall eine Verschiebung des Produktionsstarts und damit hohe Zusatzkosten.

Eine weitere Folge dieser veränderten Rahmenbedingungen wird sein, dass neben vollständigen Dauerversuchen zur Absicherung und Freigabe vermehrt umfänglich reduzierte Prüfungen oder Ersatzdauerläufe durchzuführen sind. Dieses wird besonders bei konkreten Problemen im Feld, Zeitverzögerungen in der Entwicklung oder späteren Maßnahmen zur Kostenreduzierung gelten, wenn einzelne Bauteile oder Teilsysteme abzusichern sind. Als Beispiel sei die Absicherung eines Bauteils im Türschloss genannt, bei dem zur Kosteneinsparung eine Änderung des Materials vorgesehen ist. Die übrigen Komponenten der Tür sind für die Erprobung nicht relevant. Derartige Aufgaben sind besonders zeitkritisch und erfordern ein hohes Maß an Flexibilität von der Prüfeinrichtung, da sehr kurzfristig Ergebnisse vorliegen sollen.

Das Ziel der realen Dauererprobung von Karosseriekomponenten ist eine möglichst realitätsnahe Aussage über die Haltbarkeit der späteren Serienkomponenten. Dafür werden jedoch aus Zeit- und Kostengründen für eine statistische Auswertung zu wenige Prüfungen durchgeführt. Die Übertragbarkeit der Testergebnisse auf die Serie muss dennoch so zuverlässig sein, dass mindestens für den Gewährleistungskosten- und Kulanzzeitraum sichere Prognosen über entstehende Kosten zur Behebung der Mängel möglich sind.

### **3.2.1 Voraussetzungen für die reale Erprobung**

Eine reale Erprobung von Bauteilen ist erst in der späten Phase der Fahrzeugentwicklung möglich, da mehrere Voraussetzungen erfüllt sein müssen. Die wichtigste Anforderung ist die Verfügbarkeit von prüfwürdigen Bauteilen. Bei Hardwarekomponenten sollten das Material sowie die Fertigungsmittel einen möglichst ausgereiften Stand vorweisen. Softwarekomponenten müssen bis auf die Feinabstimmung alle geplanten Funktionen mit den dazugehörigen Parametern und Kennlinien beinhalten. Für den Aufbau und die Inbetriebnahme der Prüfung sind weitere wichtige Daten, beispielsweise Anzugsmomente und Spaltmaße auf der Hardwareseite, Betriebsspannungen und Pausenzeiten für elektronische Komponenten sowie Inhalte von Busnachrichten zur Ausführung von Funktionen auf dem Steuergerät erforderlich.

Zur Erzielung von aussagekräftigen und belastbaren Ergebnissen ist ein den Prüfkomponten angepasstes und geeignetes Prüffeld notwendig. Dazu gehört insbesondere eine realitätsnahe Ansteuerung und Belastung der Komponenten sowie eine ausreichend genaue Messung mit anschließender Analyse der auftretenden Veränderungen am Prüfling. Die strikte Einhaltung dieser Anforderungen erweist sich in der Praxis mitunter als schwierig, besonders wenn Prüfverfahren und Prüfeinrichtungen für Innovationen im Fahrzeug neu entwickelt werden müssen. Neben den technischen Voraussetzungen sind detaillierte Kenntnisse über Höhe, Dauer und Zusammensetzung der späteren Belastungen auf die Fahrzeugkomponenten maßgebend. Nur dadurch ist das Ergebnis der Tests auf die zu erwartende Haltbarkeit der Serienteile übertragbar. Besonders bei Lebensdauerprüfungen ist ein ausreichendes Zeitfenster für die Prüfung vorzusehen. Ansonsten sind nötige Pausenzeiten, beispielsweise zwischen der Betätigung elektrischer Antriebe oder bei Verweilzeiten unter extremen klimatischen Bedingungen, nicht gewährleistet.

Lebensdauerprüfungen im Zeitraffer sind immer mit Kompromissen verbunden. Im Vorfeld einer Prüfung ist zu definieren, welche Kompromisse zu Gunsten der Laufzeit bewusst eingegangen werden können. Eine Variante ist die Erhöhung der Belastung bei gleichzeitiger Verringerung der Betätigungszyklen. Um eine Überhitzung elektrischer Antriebe zu verhindern, kann zur Minimierung von Pausenzeiten eine externe Kühlung, beispielsweise durch Druckluft, Abhilfe schaffen. Steht nicht das Gesamtsystem im Fokus der Erprobung, sondern werden nur einzelne Komponenten getestet, sind Beschädigungen an nicht erprobungsrelevanten Bauteilen tolerierbar. Wird beispielsweise an einem Fenster lediglich ein neues Dichtungssystem erprobt, kann der Abrieb an der Dichtung ohne Pausen in der Fensterheberbewegung nachgestellt werden. Der Austausch eines durch Überhitzung beschädigten Fensterhebermotors ist wirtschaftlich sinnvoller als minutenlange Pausen. In der Praxis sind derartige Kompromisse zwischen Entwickler, Verwender und Prüfer abzuwägen.

Eine weitere Voraussetzung für die reale Erprobung ist ein umfassender Kenntnisstand des betreuenden Ingenieurs bezüglich der zu prüfenden und der benachbarten Bauteile. Dieses gilt besonders bei elektronischen Komponenten, da durch die Fülle verschiedenster Funktionalitäten teilweise nicht zwischen gewollten Funktionen oder Fehlern unterschieden werden kann.

### **3.2.2 Potentielle Mängel am Prüfling**

Für die Entwicklung eines Prüfsystems zur Untersuchung von Langzeitbetriebsfestigkeiten technischer Produkte müssen die möglichen Fehlerarten bekannt sein. Diese können für die Fahrzeugentwicklung in vier wesentliche Gruppen aufgeteilt werden: Das Konzept einschließlich der Konstruktion bei Hardwarekomponenten, die Materialauswahl samt Fertigung, die Montage sowie der Systementwurf in Verbindung mit der Programmierung bei Softwarekomponenten. Bei der Absicherung besteht dazu eine Gefahr durch unsachgemäße Benutzung oder extreme Umweltbedingungen. Die Betriebsfestigkeitsuntersuchung setzt sich aus zwei Phasen zusammen, in denen diese Mängel deutlich werden. Vor der eigentlichen Dauererprobung kommt es im Rahmen des mechanischen Aufbaus und der elektrischen Inbetriebnahme zur ersten Funktionsprüfung. Dabei wird analysiert, ob sämtliche hardware- und softwaretechnischen Eigenschaften und Funktionen am Prüfaufbau vorhanden sind und ordnungsgemäß arbeiten. Nicht abstellbare Mängel und mögliche Auswirkungen müssen vor dem Start einer Erprobung bekannt sein, um später Fehlinterpretationen zu vermeiden. Die zweite Phase ist die eigentliche Lebensdauererprobung in Gestalt eines Dauerlaufs. Hierbei ist zu prüfen, ob mit zunehmender Alterung Eigenschaften und Funktionen ganz oder teilweise beeinträchtigt sind. Die Auswirkungen von extremen Umweltbedingungen stehen besonders im Blickpunkt. Bei auftretenden Störungen sind unverzüglich Gegenmaßnahmen einzuleiten, da die Kosten für ihre Behebung mit Blick auf den Serienanlauf immens ansteigen und im ungünstigsten Fall nach einer Auslieferung an den Kunden unkalkulierbar werden.

#### **3.2.2.1 Konzept- und Konstruktionsfehler**

Gemäß dem Produkthaftungsgesetz liegt ein Konstruktionsfehler vor, wenn das Produkt durch die zu Grunde liegende Konzeption nicht der zu erwartenden Qualität in Bezug auf Funktionalität und Sicherheit entspricht. Gründe dafür können Fehler in der Logik sowie zu schwache Dimensionierungen sein. Neben den reinen mechanischen Komponenten, wie beispielsweise Materialauswahl, Materialstärke oder Anzugsmoment, bezieht diese Fehlerart auch einige elektronische Komponenten wie Antriebe, Kabelbäume und Steckverbindungen mit ein [Erb-06]. Die Auswirkungen dieser systematischen Fehler sind oft sehr schwerwiegend und führen nicht selten zu kompletten Funktionsausfällen. Eine leichte Absenkung der Tür, bedingt durch



eine zu schwache Auslegung oder ungleiche Spaltmaße, hat zwar keine funktionskritischen Folgen, führt aber zu optischen Mängeln. Besonders bei Karosserieelementen, die im primären Blickfeld des Kunden liegen, sind die Qualitätsempfindungen getrübt. Konstruktions- und Konzeptfehler werden häufig bereits während des Aufbaus und der Inbetriebnahme einer Prüfung entdeckt, wenn beispielsweise Bauteile sichtbar kollidieren oder Bohrlöcher falsch positioniert sind. Unzureichende Dimensionierungen hingegen treten im Dauerversuch erst im Laufe der Zeit, bei Steifigkeitsprüfungen oder bei Crashversuchen in Erscheinung. Bezüglich möglicher Konstruktions- und Konzeptfehler erwartet der Kunde vom Hersteller alle erdenklichen Vorkehrungen, so dass bei der Verwendung des Produkts kein Schaden entsteht.

### **Auswirkungen von Extrembelastungen**

Die Karosserie eines Taxis oder Lieferwagens muss ungleich höhere Belastungen ohne Beschädigungen überstehen, als dies bei der durchschnittlichen Nutzung eines Kraftfahrzeugs der Fall ist. Die allgemeine Auslegung der Konstruktionen auf diese Extrembeanspruchungen ist aus Kostengründen nicht möglich. Trotzdem muss eine Abwägung zwischen tolerierbaren Bauteilveränderungen und den Extrembelastungen geschuldeten Systemausfällen stattfinden. Letztere dürfen in keinem Fall auftreten. Ein Teil der Dauererprobung besteht daher aus Belastungen, die über das *normale Maß* hinausgehen. Theoretisch hat der Kunde unbegrenzte Möglichkeiten, ein Fahrzeug bewusst oder unbewusst unsachgemäß zu behandeln, wie das Verpolen der Batterie, das Aufstützen auf eine geöffnete Tür oder eine Überschreitung der maximalen Zuladung [Hau-04].

#### **3.2.2.2 Material- und Fertigungsfehler**

Material- und Fertigungsfehler sind nach dem Produkthaftungsgesetz Mängel, die im Einzelfall zufällig während der Produktion auftreten. Gründe dafür können Schwankungen in der Qualität des angelieferten Materials sowie Defekte an einer Produktionsmaschine, beispielsweise an einer Schweißzange, sein [Erb-06]. Treten während der Entwicklung vermeintliche Fehler dieser Art auf, ist der Fehlertyp genau zu analysieren, da Konstruktions- oder Konzeptfehler im Laufe der Entwicklung viel wahrscheinlicher sind. Materialfehler werden in dieser Phase eher auf eine unzureichende Dimensionierung als auf eine zufällige Fehlstelle im Werkstoff zurückgeführt. Die entwicklungsbegleitende Erprobung kann aufgrund der viel zu geringen Anzahl von Prüflingen nicht gezielt auf diese Fehlerart eingehen. Während des gesamten Produktionszeitraums ist jedoch eine strikte Überwachung zur Sicherung der Qualität durchzuführen. Die Verantwortung wird von der Entwicklungsabteilung auf das Produktionswerk übertragen.

#### **3.2.2.3 Montagefehler**

Als Montagefehler werden Abweichungen vom Sollzustand bei zusammenmontierten Komponenten definiert. Diese können durch menschliche oder maschinenbedingte Ursachen entstehen. Beispiele sind eine schief eingedrehte oder mit falschem Anzugsmoment befestigte Schraube, eine fehlende Verclipsung zwischen der Verkleidung und dem Blech oder ein unsauber verlegter Bowdenzug. Fehler dieser Art können sowohl beim Aufbau von Prototypen für die Erprobung als auch in der späteren Produktion entstehen. Ein zentrales Problem ist, dass die genannten Fehler während der Dauererprobung nur erkannt werden, wenn diese sehr offensichtlich sind oder Funktionsbeeinträchtigungen zur Folge haben. Als Beispiel sei genannt, wenn bei einem nicht korrekt verlegten Bowdenzug das Seil durchscheuert und sich

dadurch das Türschloss nicht mehr entriegeln lässt. Eine direkte Überprüfung der Montage kann während einer Dauererprobung nur in der Phase des Aufbaus und der Inbetriebnahme erfolgen. Bei später auftretenden Fehlfunktionen ist zu analysieren, ob Montagefehler eine mögliche Ursache darstellen. Beim Vorliegen eines Montagefehlers sind die betroffenen Bauteile derart zu optimieren, dass keine erneuten Mängel zu erwarten sind.

#### **3.2.2.4 Softwarefehler**

Der Trend, Hardwarefunktionen durch Software zu ersetzen und zu erweitern, nimmt zu. Anders als mechanische oder elektrische Probleme basieren Softwarefehler nicht auf physikalischen oder chemischen Ursachen, sondern auf Fehlern in der Vollständigkeit oder Logik eines Programmcodes. Softwarefehler weisen allgemein ein systematisches Verhalten auf, im Zusammenspiel mit Hardwarekomponenten in mechatronischen Systemen können die Fehler wiederum zufällig auftreten [Pos-04], [Rei-08], [Ber-09].

Für die Dauererprobung von Karosserieelementen ist die Aufteilung der Software auf dem Steuergerät in zwei Bereiche zweckmäßig. Ein Teil ist der Programmcodes. Die hier enthaltenen Funktionen sind zuständig für die Verarbeitung von Sensorsignalen, die Ansteuerung von Aktoren und die Kommunikation mit anderen Steuergeräten im Fahrzeug über Bussysteme. Fehler im Programmcodes werden sehr frühzeitig bei der Inbetriebnahme des Dauerlaufs erkannt, weil benötigte Funktionen nicht ausführbar sind. Die Ursache ist überwiegend darin zu finden, dass die Funktionen zum aktuellen Entwicklungsstand noch nicht fertig programmiert oder implementiert sind. Der zweite Teil besteht aus Parametern und Kennlinien, die sehr fein auf den Fahrzeugtyp oder die Variante abgestimmt sein müssen. Auf Basis dieser Daten wird in weiterer Abhängigkeit von beispielsweise der Außentemperatur oder der Fahrzeuggeschwindigkeit die Funktionsweise des Steuergeräts beeinflusst. Fehler in der Parametrierung treten im Verlauf der Prüfung erfahrungsgemäß im Zusammenhang mit extremen Umweltbedingungen in Erscheinung. Der Einklemmschutz am Fensterheber ist eine Sicherheitsfunktion mit großer Abhängigkeit von der Außentemperatur. In der Praxis haben fehlende oder nicht abgestimmte Parameter bei sehr kalten Umgebungstemperaturen und bedingt durch die Schwergängigkeit der Scheibe zur Folge, dass permanent ein Einklemmfall detektiert wird und dadurch die Scheibe reversiert.

Steuergeräte besitzen die Eigenschaft, sich selbst sowie die angeschlossene Sensorik und Aktorik während des Betriebs fortlaufend zu überwachen und auftretende Fehler im sogenannten Fehlerspeicher abzulegen. Dieser Fehlerspeicher ist oftmals sehr nützlich, um in der Phase der Inbetriebnahme die Ursache von Funktionsstörungen aller Art zu finden. Während der Dauererprobung kann über Einträge im Fehlerspeicher der Einfluss der Alterung von Hardwarekomponenten überwacht werden. Softwarefehler sind im Allgemeinen zügiger korrigierbar als Hardwarefehler, dies gilt aber aus organisatorischen Gründen nicht für den Programmcodes auf Steuergeräten im Umfeld der Erprobung. Daher müssen für eine erfolgreiche Durchführung der Prüfung in der Praxis häufig kurzfristige Änderungen am Prüfstand vorgenommen werden.

#### **3.2.2.5 Sonstige Fehler**

Neben den bis hierher aufgeführten in der Technik begründeten Fehlern gibt es auch andere Gründe, die während der Durchführung von Betriebsfestigkeitsuntersuchungen zu Problemen führen können. Bei der Entwicklung von mechatronischen Komponenten, beispielsweise einer Fahrzeugtür, sind eine Vielzahl von Personen verschiedenster Fachstellen beim Fahrzeugher-

steller, Ingenieurdienstleister und Zulieferbetrieb involviert. Eine mangelnde Kommunikation und Abstimmung führt zu einer allgemein unzureichenden Kenntnis über die Eigenschaften und Funktionen des Prüflings samt den nötigen Einsatzbedingungen. Die Folge sind unangemessene Reaktionen und Schlussfolgerungen bei auftretenden Fehlern. Besonders bei tiefgreifenden Systemänderungen sind alle beteiligten Fachstellen zu informieren.

### 3.2.3 Prüfvorschriften

Prüfvorschriften basieren auf Normen und Vorschriften, auf deren Grundlage die Erprobung von Bauteilen durchzuführen ist. Sie beinhalten Informationen über die allgemeinen Prüfbedingungen, den Prüfling und dessen Aufbau sowie Einstellungen. Dazu kommen Umweltbedingungen, Messungen und zu verwendende Messmittel, der detaillierte Prüfablauf und Angaben zum Erscheinungsbild und Inhalt der Dokumentation. Die Prüfvorschrift ist für den Entwickler ein verbindlicher Teil des Lastenhefts und kommt bei Freigabeuntersuchungen von Neuentwicklungen und Zulieferteilen sowie bei Änderungen gegenüber dem aktuellen Freigabestand zur Anwendung. Sie ist ein Dokument, welches durch Produktinnovationen, neue Fehlerbilder, wirtschaftliche Interessen oder gesetzliche Änderungen ständig beeinflusst und aktualisiert wird. Der Verwender ist daher verpflichtet, sich vor einer Prüfung über den aktuell gültigen Stand zu informieren.

Trotzdem bedarf es einer ständigen Diskussion, ob in einer Prüfvorschrift festgelegte Abläufe das tatsächliche Kundenverhalten oder die Bedingungen im Feld hinreichend abbilden. Bei manchen Prüfungen können definierte Messungen überflüssig sein, andere Messungen wiederum wären sinnvoll und würden zusätzlich wichtige Informationen liefern. Nicht jede wünschenswerte Prüfung ist in der Realität wegen zeitlicher, finanzieller oder physikalischer Grenzen durchführbar. Manche Prüfabläufe sind auch in nicht optimaler Form akzeptiert, da ansonsten eine Prüfung erst gar nicht realisierbar wäre.

Erscheint eine Prüfmethode als veraltet, weil beispielsweise die Messergebnisse trotz intaktem Messequipment unrealistisch wirken oder die Beanspruchung realitätsfern ist, steht oftmals die Vergleichbarkeit mit Prüfungen aus Vorgängerprojekten im Vordergrund. Hat ein Bauteil mit vergleichbaren Prüfergebnissen in der Serie keine Probleme bereitet, kann die Freigabe auch auf Basis dieser Vergleichstests erfolgen.

Qualität lässt sich in vielen Fällen nicht durch objektive Größen, sondern nur durch subjektives Empfinden, wie Klangbild, optische Erscheinung oder Anfühlen, ermessen. Prüfvorschriften versuchen soweit wie möglich, diesen subjektiven Empfindungen objektive Größen zuzuordnen. Diese Herausforderung ist umso größer, je häufiger das Personal wechselt, welches die Beurteilung vornimmt.

Prüfvorschriften lassen sich auch zur Beeinflussung des Wettbewerbs instrumentalisieren, wenn bestimmte Alleinstellungsmerkmale eines Anbieters als verpflichtende Bestandteile von Prüfungen aufgenommen werden.

Jeder Fahrzeughersteller und Zulieferer hat bei der Prüfung von Karosseriekomponenten eine eigene Philosophie. Einige Hersteller testen jedes Bauteil einzeln, andere wiederum prüfen alle betroffenen Bauteile eines Systems in Kombination, manche setzen auf Statistik, wieder andere führen nur Einzelprüfungen durch. Auch die Anzahl und Ausgestaltung von Testzyklen sind verschieden. Das in diesem Projekt entwickelte Prüfsystem muss in der Lage sein, die beschriebenen Anforderungen zu erfüllen.

### 3.2.4 Prüfstände für Karosseriekomponenten

Der Anteil von elektronischen Komponenten hat auch im Bereich der Karosserie zugenommen. Bei der Betrachtung von Prüfständen zur Absicherung von Karosserieteilen entsteht jedoch der Eindruck, dass sich die Prüftechnik nicht in angemessener Weise weiterentwickelt hat. Prüfstände für Karosseriekomponenten sind grundsätzlich der Gruppe von *Einzellösungen* zuzuordnen. Bei Dienstleistungsunternehmen werden in jeder Niederlassung individuelle Lösungen eingesetzt, da die Prüfvorschriften der dortigen Fahrzeughersteller sehr differieren und Prüfstände mangels Flexibilität nicht variabel verwendbar sind.

Im Jahr 2007 beinhaltete der Testablauf überwiegend die Ausführung mechanischer Betätigungen, Vorrichtungen dafür waren speziell angefertigte Gestelle aus Profilschienen. Für die Bewegungen sorgten Servomotoren oder pneumatische Zylinder. Eine kundennahe Betätigung von Türen und Klappen konnte damit nur bedingt nachgestellt werden. Besonders unter extremen klimatischen Belastungen war die Einstellung der vorgegebenen Betätigungsgeschwindigkeiten sehr zeitaufwendig. Pneumatische Antriebe zeigten hier ein sehr ruckartiges Beschleunigungsverhalten, als Folge traten untypische Stöße und Schläge auf. Bedingt durch diese schwer zu kontrollierenden Eigenschaften war eine Reproduzierbarkeit der Belastung nicht gegeben. Die Rüstzeiten betrugen durch notwendige Anpassungen bei jedem neuen Dauerlauf in der Regel mehrere Wochen.



Abbildung 11: Dauerlaufeinrichtungen zur Türerprobung

Eine schnell und flexibel programmierbare Schnittstelle zur Fahrzeugelektronik wurde nicht eingesetzt. Lediglich einzelne in der Anzahl begrenzte Botschaften auf den Fahrzeugbussen konnten aus SPS – Signalen erzeugt oder ausgewertet werden. Eine Restbussimulation zur Schaffung einer realitätsnahen Betriebsumgebung für die Steuergeräte war nicht vorgesehen. Die Erfassung, Auswertung und Analyse von Messdaten erfolgte nicht in der Intensität, dass dadurch eine umfangreiche Unterstützung bei der Bewertung der Ergebnisse gegeben war. Messungen wurden nur in einer so geringen Anzahl während der Prüfung durchgeführt, dass daraus keine statistischen Aussagen abgeleitet werden konnten. Durch den geringen Automatisierungsgrad waren die Messungen vielen äußeren Einflüssen ausgesetzt, besonders durch oft wechselnde Mitarbeiter. Beispiele sind das Ermitteln des Auslösewegs am Türaußengriff mittels eines Lineals oder die Messung der Rastkräfte einer Türbremse per Handkraftmessdose. Da der Dauerlauf im Idealfall 70 Prozent außerhalb der betriebsüblichen Ar-

beitszeit erfolgt, ist besonders die Kontinuität in der manuellen Messdatenerfassung erschwert. Manuelle Messungen am Prüfling unter extrem kalten Bedingungen führen schon bei kurzen Öffnungen der Klimakammer zum Betreten und Verlassen zu Vereisungen am Fahrzeug. Automatische Dokumentations- und Benachrichtigungsfunktionen gab es nur in sehr rudimentärer Ausführung. Dadurch musste eine Prüfung permanent durch Mitarbeiter überwacht werden. Vergleichbare Beschreibungen finden sich in den Quellen [Ter-07], [Vog-07b] und [Iav-08].

### 3.3 Robotik

Überall dort, wo der Einsatz von Robotern lohnenswert erscheint und sich nach kurzer Zeit amortisiert, ist bereits eine Sättigung festzustellen. Ein Schwerpunkt der universitären Forschung ist es daher, den Einsatz von Robotertechnik in Anwendungsgebieten mit noch geringer Verbreitung gewinnbringend zu ermöglichen. Speziell bei sinkenden Seriengrößen weisen Industrieroboter gegenüber starr automatisierten Anlagen wirtschaftliche Vorteile auf, kleine und mittelständische Unternehmen sollen dadurch am Markt konkurrenzfähig bleiben. Ausbaufähige Einsatzgebiete sind beispielsweise flexible Produktionsanlagen, die Medizin- oder die Messtechnik [Vog-08].

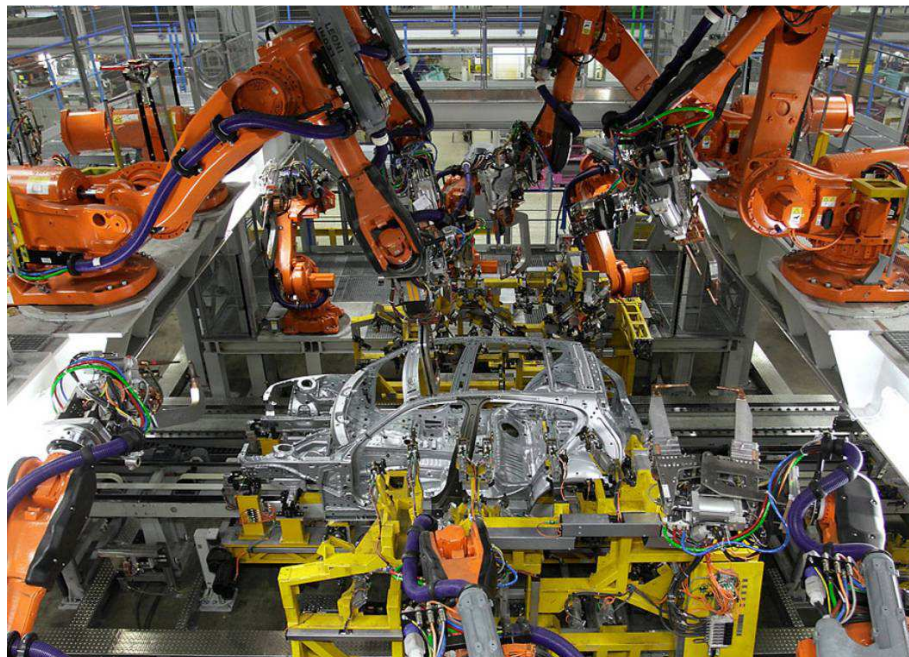


Abbildung 12: Roboter in der Automobilproduktion [Bmw-11]

Roboter wurden in erster Linie für die Automobilindustrie entwickelt, um dort bei Schweiß- und Lackieranwendungen die Produktivität zu steigern und die Mitarbeiter zu entlasten. Durch den Zuwachs an Leistungsfähigkeit bei den Robotercontrollern und die Möglichkeit zur Verarbeitung von zusätzlichen Sensorsignalen eröffnen sich neue Einsatzgebiete. Für die Zukunft werden dadurch erhebliche Wachstumspotentiale im Bereich der Robotik prognostiziert [Zab-93], [Bau-09].

Im Verlauf dieser Arbeit wird eine *aufgabenorientierte Schnittstelle* entwickelt, die den Einsatz von Robotern im Umfeld der Komponentenerprobung mit Blick auf technische und wirtschaftliche Gesichtspunkte ermöglicht. Das Hauptaugenmerk liegt dabei auf der Integration



von bereichstypischer Messhardware und einer anwendungsorientierten Programmierungsumgebung für Versuchingenieure. Die zum Verständnis dieser Schnittstelle benötigten Grundlagen werden in diesem Kapitel zusammengefasst.

### 3.3.1 Robotertypen

Eine zweckmäßige Einteilung der heute überwiegend eingesetzten Robotertypen erfolgt durch die mechanischen Grundkomponenten, das Erscheinungsbild sowie den primären Einsatzbereich.

#### 3.3.1.1 Industrieroboter

In der *DIN EN ISO 8373* sind Industrieroboter als automatisch gesteuerte, bezüglich ihrer Bewegungsfolgen und Wege frei programmierbare Mehrzweck – Manipulatoren zur Handhabung von Werkzeugen oder Werkstücken in mindestens drei Achsen definiert. Die Verwendung in der Fabrikautomatisierung erfolgt in statischer oder mobiler Installation. In dieser Eigenschaft stellen Industrieroboter den ältesten und bisweilen noch größten Anteil auf dem Robotikmarkt [Vog-08]. Der Einsatz rentiert sich, da Industrieroboter schnell und präzise pausenlos tonnenschwere Lasten bewegen können und Mitarbeiter von unangenehmen und schweren Arbeiten befreien. Durch die Verwendung spezieller Schutzhüllen ist der Einsatz im Reinraum, in der Lebensmittelverarbeitung, in stark verschmutzten Umgebungen oder unter extremen Klimabedingungen möglich. Die Haupttätigkeiten sind Montage-, Schweiß-, Klebe- und Beschichtungsvorgänge. Die Sparte der Industrieroboter kann durch den jeweiligen Aufbau der Kinematik des Arms und die Art der Ansteuerung weiter unterteilt werden. Die mögliche Bewegungsfreiheit und Reichweite des Roboterarms wird durch die Anzahl, Art und Anordnung der Achsen sowie die dazwischen befindlichen Verbindungselemente beeinflusst. Gelenke unterscheiden sich in Drehachsen und Schubachsen, diese können in einer Reihe, parallel oder in einer Kombination angeordnet sein.

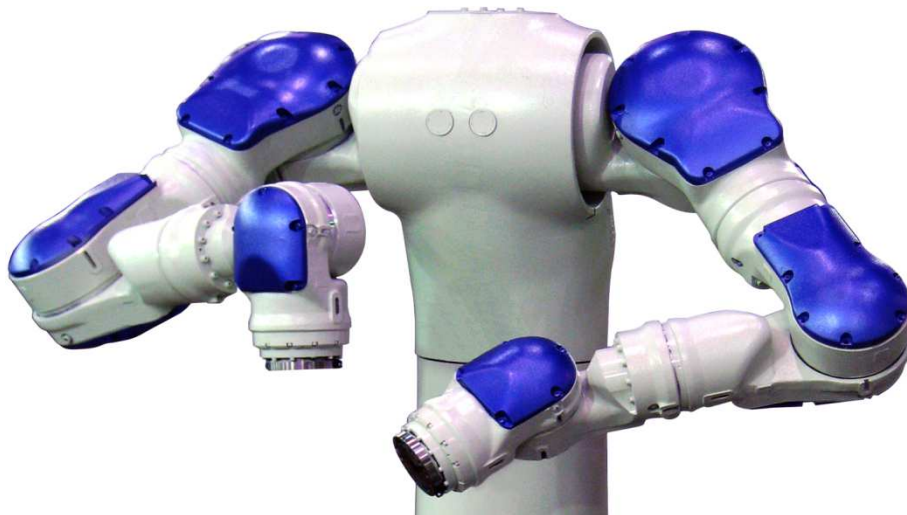


Abbildung 13: Roboterarm des Motoman SDA10D mit 15 Achsen [Mot-10]

Typische Armvarianten sind vierachsige SCARA – Roboter für schnelle *Pick – and – Place – Anwendungen* oder sechssachsige dreidimensional bewegliche Knickarmroboter für den flexiblen Einsatz bei Bearbeitungs-, Montage- und Handhabungstätigkeiten. Für spezielle Verwendungen gibt es darüber hinaus Roboter mit bis zu 15 Achsen. Ein Beispiel ist der Dual-

armroboter *Motoman SDA10D* mit Bewegungsfreiheiten, die einem menschlichen Oberkörper entsprechen.

An der *Technischen Universität Bergakademie Freiberg* wurde eine Methode zur Koordinatentransformation und Bahninterpolation des neunachsigen Roboterarms *MRC 9-15* der Firma *Maschinenbau Mauersberger* entwickelt. Dieser besitzt im Vergleich mit sechachsigen Knickarmrobotern zusätzlich eine Drehachse und zwei Schubachsen in einer offenen kinematischen Kette. Die Erhöhung der Flexibilität bei Bewegungsabläufen führt im Gegenzug zu einer dreifach redundanten Kinematik. Mit Hilfe der entwickelten Methode ist eine Ansteuerung des Roboterarms in Echtzeit möglich, ohne dabei die Vorteile der Zusatzachsen einzubüßen [Löb-06].

Neben dem Arm bestehen Roboter aus einem Controller, welcher mittels eines Rechners die Bahnkurven generiert und zeitgleich die nötigen Ansteuerungen der Motoren zum Erreichen der Zielposition vornimmt. Die Rechengeschwindigkeit der Controller und die Verfügbarkeit von Schnittstellen zur Kommunikation mit externen Geräten haben sich in den vergangenen Jahren stark erhöht. Heute sind Roboter in der Lage, Informationen von zusätzlichen externen Sensoren in die Bewegungsberechnung mit einzubeziehen.

Industrieroboter sind durch ihre große Verbreitung in den vergangenen Jahren zu Standardkomponenten geworden, der Preis dafür ist stark gefallen [Vog-08]. Je nach Bedürfnis kann das passende Modell aus einer Vielzahl am Markt verfügbarer Geräte verschiedenster Hersteller ausgewählt werden. Ersatzteile sind in der Regel ab Lager lieferbar, maschinenbedingte Stillstände sind daher minimal.

### 3.3.1.2 Serviceroboter

Als Serviceroboter gelten voll- oder teilautonome Maschinen, die außerhalb der industriellen Praxis den Menschen bei der Ausführung von Tätigkeiten des täglichen Lebens unterstützen. Im Gegensatz zu Industrierobotern sind diese in der Regel für den mobilen Einsatz konzipiert und hinsichtlich der Bedienung auch für ungeschulte Bediener einsetzbar [Hau-07], [Vog-08]. Ihr Einsatzgebiet erstreckt sich über professionelle Anwendungen, wie beispielsweise die Fassadenreinigung von Hochhäusern, die Flugzeugreinigung, Unterwasseroperationen oder Melkanlagen für Kühe. Auch als Konsumgüter wie Rasenmäher, Tennisballsammler oder universelle Haushaltshilfen steigt die Verbreitung im privaten Umfeld stetig [Lin-09]. Für eine Orientierung und Arbeitsraumüberwachung zum Zweck einer Kollisionsvermeidung verfügen derartige Systeme über optische und akustische Sensoren und meistens zusätzlich über eine schnurlose Kommunikationsschnittstelle mit einer Basisstation. Zur Sicherung der Energieversorgung müssen kabellose Systeme frühzeitig die Basisstation zum Aufladen der Akkumulatoren anfahren [Ste-06].

### 3.3.1.3 Humanoide Roboter

Humanoide Roboter sind in erster Linie durch ein menschenähnliches Aussehen und Verhalten gekennzeichnet. Sie sollen in der Lage sein, Aufgaben selbständig zu planen, auszuführen und zu überwachen. Da die Interaktion identisch ist mit der zwischenmenschlichen Interaktion, ist für die Bedienung keine besondere Anleitung nötig. Dieses Forschungsfeld ist das jüngste im Bereich der Robotik und legt seinen Schwerpunkt auf die Lernfähigkeit der Roboter, die Mensch – Roboter – Kooperation, die Sensordatenauswertung und die modellhafte Abbildung des Menschen und seiner Umwelt [Wör-03]. Humanoide Roboter sind eine weitere

Entwicklungsstufe der Serviceroboter und können in engerer Kooperation mit dem Menschen arbeiten. Ihre Körper sind dabei auf die menschliche Umgebung abgestimmt und ihre Bewegungsmuster gut einschätzbar [Hau-07].



Abbildung 14: Humanoider Roboter als Helfer auf der Baustelle [Boc-04]

#### 3.3.1.4 Visionen der Roboterforschung

Industrie- und mittlerweile auch Serviceroboter haben heute durch die flächendeckende Verbreitung eine sehr große wirtschaftliche Bedeutung. Humanoide Roboter befinden sich noch im Stadium der Grundlagenforschung. Im Rahmen von Forschungsprojekten wird an der Realisierung zukünftiger Schlüsselqualifikationen gearbeitet. Dazu gehören multimodale Dialoge, die Situationserfassung und Situationsbewertung, das Verständnis menschlicher Handlungen, autonomes Entscheiden und Problemlösen, die robusten Anwendungen im Alltag und in öffentlichen Bereichen, Gedächtnisstrukturen sowie eine Selbstüberwachung [Dil-06], [Pir-08].

Sollten diese Visionen in der Zukunft tatsächlich zur realitätsnahen Verwendbarkeit reifen, ergäben sich für das Versuchswesen ganz neue Möglichkeiten. Humanoide Roboter könnten nach kurzer Einweisung oder Beobachtung der menschlichen Handlung selbständig Dauerversuche ausführen und dabei die Vorteile von Mensch und Maschine vereinen. Bei aller berechtigten Euphorie sollte aber der Blick für die Realität nicht verloren gehen. Die für uns Menschen selbstverständlichen Bewegungsabläufe, wie beispielsweise das Zuziehen einer Fahrzeugschleuse am Türinnengriff, sind durch Roboter nur mit viel Geschick und Erfahrung umsetzbar. Der menschliche Bewegungsapparat ist ein in vielen Millionen Jahren entwickeltes Wunderwerk, welches nur sehr schwer zu kopieren ist [Hau-07].

#### 3.3.2 Roboterwerkzeuge

Roboterwerkzeuge bilden die mechanische Schnittstelle zwischen dem Roboter und dem zu bearbeitenden oder bewegenden Gegenstand. Sie werden am standardisierten Flansch der



letzten Roboterachse befestigt und sind in der Regel speziell für die zu verrichtende Aufgabe entwickelt. Häufig verwendete Werkzeuge bei Industrierobotern sind Schweißzangen, Greifer, Lackierpistolen und Werkzeuge für Bearbeitungs- oder Montageaufgaben. Da die Traglast der Roboter eingeschränkt ist, sind die Masse und deren Anordnung möglichst optimal auszulegen. Roboterwerkzeuge haben in vielen Fällen Messtechnik integriert, um beispielsweise Kräfte zu überwachen, Barcodes mit Hilfe von Kameras einzulesen oder den Roboterarm für eine hohe Präzision nach einer bestimmten Anzahl von Zyklen automatisch zu kalibrieren. Werden bei einem Arbeitsgang mehrere Werkzeuge benötigt, kann mittels eines Wechselsystems ein automatischer Werkzeugwechsel stattfinden.

Ein Forschungsgebiet ist die Entwicklung eines möglichst universell einsetzbaren Roboterwerkzeugs in Form der Nachbildung einer menschlichen Hand. Eine menschliche Hand verfügt über 23 verschiedene Freiheitsgrade, ein stufenloses Feingefühl sowie zirka 17.000 Sensoren für Druck, Temperatur und Berührungen. Sie ist dafür ausgelegt, bei einem Eigengewicht von durchschnittlich nur 600 Gramm wesentlich schwerere Massen greifen und sicher halten zu können. Aktuelle Forschungsergebnisse weisen bereits sehr viele Übereinstimmungen mit dem Original auf [Dil-06], [Hau-07], [Joo-07], [Gre-10].

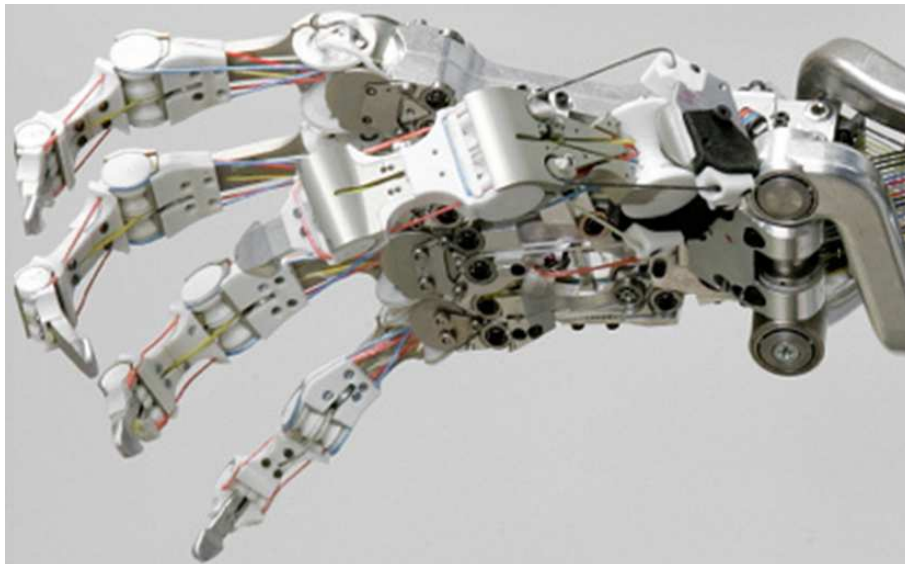


Abbildung 15: Roboterhand des DLR [Gre-10]

### 3.3.3 Koordinatensysteme

Koordinatensysteme dienen im Zusammenhang mit Robotern als eindeutige Bezugsgrundlage für weitere Koordinatensysteme und Punkte eines Bewegungsablaufs im dreidimensionalen Raum. Der Bediener kann diese je nach Bedarf und persönlichen Vorlieben bei der Roboterprogrammierung verwenden und zwischen ihnen umschalten. Da die Bezeichnungen zwischen den Roboterherstellern variieren können, sind für den weiteren Verlauf dieser Arbeit folgende Bezeichnungen festgelegt.

Das *Weltkoordinatensystem* ist fest mit der Umwelt des Roboters verbunden und fungiert als oberste Referenz für geometrische Zusammenhänge, besonders bei Anlagen mit mehreren Roboterarmen. Das *Roboterkoordinatensystem* ist fest mit dem Sockel eines Roboters verbunden. Der Ursprung liegt üblicherweise auf der Drehachse der Achse 1. Bei Anlagen mit nur einem Roboter sind das Welt- und das Roboterkoordinatensystem in der Regel identisch.

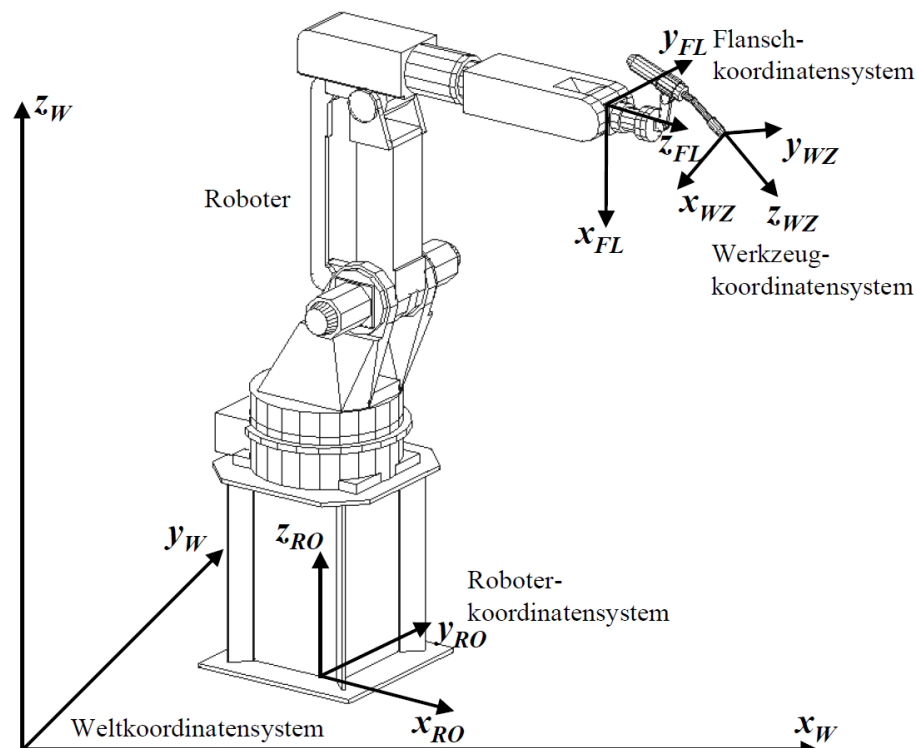


Abbildung 16: Koordinatensysteme an einem Industrieroboter Roboter [Löb-07]

Am Ende der seriellen kinematischen Kette ist am Flansch das gleichnamige *Flanschkoordinatensystem* definiert und ebenfalls fest mit dem Roboterarm verbunden. Es befindet sich an der Schnittstelle zum Roboterwerkzeug. Das *Werkzeugkoordinatensystem* ist in zweckmäßiger Ausrichtung direkt am Arbeitspunkt des Werkzeugs platziert. Der Ursprung des Werkzeugkoordinatensystems wird auch als *Tool Center Point (TCP)* bezeichnet. Die Bewegungsrichtung  $z_{WZ}$  ist die sogenannte *Stoßrichtung*. Letztgenanntes Koordinatensystem erleichtert die manuelle Bewegung des Werkzeugs beim Teachen der Punkte, da es immer identisch zur Werkzeugausrichtung positioniert ist und damit der menschlichen Arbeitsweise und Sicht auf das Werkzeug sehr entgegen kommt.

Als weitere Bezugsgrundlage ist das *Bauteilkoordinatensystem* zu nennen, welches fest an ein vom Roboter zu bearbeitendes Bauteil gebunden ist. Die Verwendung dieses Koordinatensystems ist sinnvoll, wenn sich die Lage des Bauteils zum Roboter verändert, nicht aber die Position der Punkte am Bauteil. Weitere Beispiele sind die im Verlauf dieser Arbeit eingeführten *Tür-* und *Fahrzeugkoordinatensysteme* [Löb-07], [Web-09].

### 3.3.4 Roboterprogrammierung

Die Mechanik des Roboterarms ist für den universellen Einsatz konzipiert. Erst durch die Programmierung erfolgt die Anpassung an konkret zu verrichtende Aufgaben. Die Programmierung von Industrierobotern erfordert neben technischem Grundlagenwissen über allgemeine Programmstrukturen, Koordinatensysteme, Kommunikationsschnittstellen, robotikspezifische Datentypen und Bewegungsfunktionen ein hohes Maß an Erfahrung. Durch steigende Funktionsumfänge erhöht sich der Komplexitätsgrad von Programmierungsumgebungen stetig.

In dieser Arbeit wird als ein wesentlicher Bestandteil eine aufgabenorientierte Schnittstelle zur Programmierung von Industrierobotern entwickelt, die speziell auf die Rahmenbedingungen im Versuchswesen zugeschnitten ist. Einleitend erfolgt daher eine Übersicht der gängigen

Programmierverfahren, aus denen sich diese Schnittstelle zusammensetzt. Grundsätzlich lassen sich diese in Online- und Offline – Verfahren sowie in einer Kombination beider als hybride Verfahren einteilen. Weitere Möglichkeiten der Differenzierung sowie alternative Programmierverfahren sind in den Quellen [Kug-99], [Hau-07], [Hei-07], [Bre-08b] und [Web-09] beschrieben.

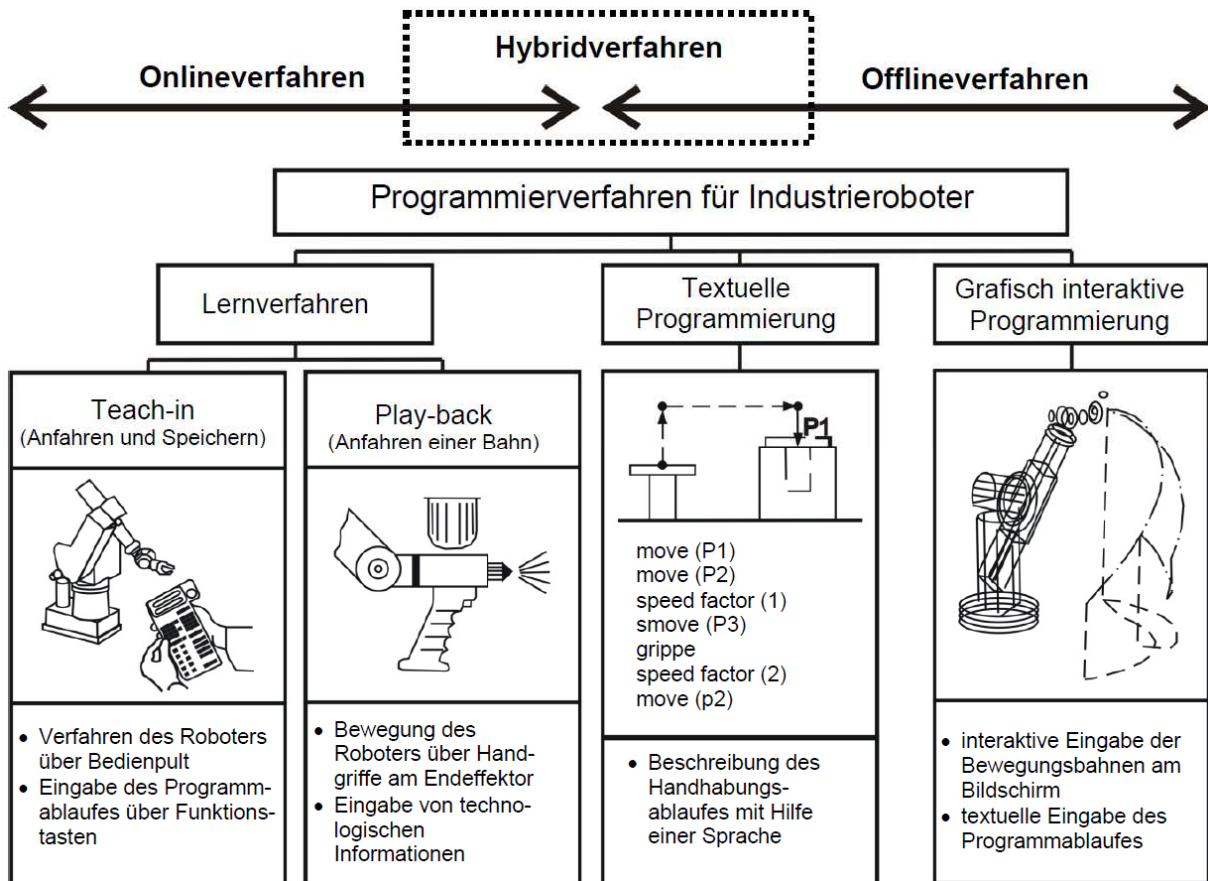


Abbildung 17: Programmierverfahren für Industrieroboter [Fri-10]

### 3.3.4.1 Online – Programmierung

Bei der Online – Programmierung erfolgt die Erstellung der Programme am und mit dem Roboter. Ein Vorteil dieser Methode liegt darin, dass die programmierten Positionen direkt mit dem Arm angefahren werden und dadurch die Bahnkurve sofort ersichtlich ist. Nachträgliche Optimierungen sind daher relativ selten notwendig und nur mit minimalem Aufwand verbunden. Als primärer Nachteil gilt, dass der Roboter während der Programmierung nicht anderweitig zu nutzen ist.

#### 3.3.4.1.1 Teach – In – Verfahren

Beim Teach – In – Verfahren positioniert der Programmierer den Roboterarm mit Hilfe des Handbediengeräts nacheinander an die verschiedenen Positionen der Roboterbahn und speichert diese im Controller. Im Anschluss werden diese Positionen automatisch abgefahren. Für die Bewegungen zwischen den programmierten Positionen sind Parameter wie Geschwindigkeit, Beschleunigung, Interpolationsart oder das zu verwendende Roboterwerkzeug zu definieren. Für die manuelle Positionierung des Roboterarms stehen verschiedene Bezugskoordinatensysteme zur Verfügung. Je nach Anwendung eignen sich die Bewegung nur einer einzel-

nen Achse, die Bewegung im Koordinatensystem des Werkzeugs oder eine Bewegung im Roboterkoordinatensystem. Alternativ zum Handbediengerät gibt es weitere Optionen zur manuellen Bewegung des Roboterarms, Beispiele sind in den Quellen [Mül-98], [Bat-10] und [Mrk-10] aufgeführt.

Die Grundsätze des Teach – In – Verfahrens, besonders das manuelle Bewegen des Roboterarms mit dem Handbediengerät, sind in relativ kurzer Zeit beherrschbar. Für das Einlernen von komplexen Bewegungsabläufen, wie das Öffnen und Schließen einer Fahrzeugtür, sind Erfahrungen in den Bereichen Singularitäten, geschwindigkeitsoptimierte Bewegungen und vorausschauende Achsstellungen nötig. Für die Generierung einer Türbewegung benötigt ein unerfahrener Programmierer mindestens einen Arbeitstag. Besonders bei der Einrichtung von Robotern im Fahrzeuginnenraum werden die engen Platzverhältnisse und Winkelbegrenzungen der Achsen schnell zur großen Herausforderung. Erfahrene Bediener hingegen benötigen für die Erstellung eines reibungslosen Bewegungsablaufs ungefähr zwei Stunden. Ein Spezialist empfindet diesen Vorgang in der Praxis häufig als uninteressant und ermüdend, trotzdem wird eine hohe fachliche Qualifikation benötigt [Pir-08].

#### **3.3.4.1.2      Playback – Verfahren**

Das Playback – Verfahren ist dadurch gekennzeichnet, dass der Roboterarm in der Nähe des Werkzeugs durch den Programmierer geführt wird. Die Bewegung wird dabei aufgezeichnet und anschließend wieder abgespielt. Bei schweren Robotern oder kritischen Umgebungsbedingungen kann der Roboterarm durch ein kinematisch vergleichbares Ersatzsystem, beispielsweise einen Miniaturroboter oder eine 6D – Mouse, geführt werden [Web-09]. Dieses Verfahren ist für unerfahrene Bediener sehr anschaulich und dadurch leicht erlernbar. Die Methode bietet sich für genau abzufahrende Bahnkurven und Geschwindigkeitsprofile an, da die von Hand geführte Kurve automatisiert abgefahren wird und dabei lediglich unter dem Einfluss der Wiederholgenauigkeit des Roboters steht. Zum Einsatz kommt diese Methode überwiegend bei Schweiß- oder Lackierrobotern. Im Gegensatz zum Teach – In – Verfahren ist das Abändern einzelner Positionsdaten nicht möglich, nötige Optimierungen an der Bewegung haben oftmals eine Wiederholung des gesamten Einlernvorgangs zur Folge [Bre-08b].

#### **3.3.4.2      Offline – Programmierung**

Zur Offline – Programmierung ist während der Programmerstellung kein Roboter nötig, da diese an einem eigenständigen Rechner ausgeführt wird. Die Programmierung kann daher schon eine Zeit lang vor dem Aufbau der Roboterzelle erfolgen. Tests und Simulation des erstellten Programmablaufs sind ebenso Bestandteil der Offline – Programmierung, soweit die technischen Gegebenheiten dies beim konkreten Projekt zulassen.

##### **3.3.4.2.1      Textbasierte Verfahren**

Bei der textbasierten Programmierung wird das Roboterprogramm durch hochsprachentypische Bestandteile einer Programmiersprache erstellt. Dazu zählen Verzweigungen, Schleifen, Sprunganweisungen und Programmaufrufe. Hinzu kommen Elemente für die Ausführung von robotertypischen Handlungen, wie Bewegungsbefehle, Bewegungsparameter und entsprechend angepasste Datentypen. Unterstützend stehen Editorfunktionen, wie Syntaxcheck, Kopieren und Einfügen, automatisches Einrücken und Wortvervollständigungen zu Verfügung.

Bei dieser Art der Programmierung sind Änderungen oder Kommentare zur Dokumentation über die Standardtastatur eines Rechners sehr komfortabel möglich.

#### **3.3.4.2.2 Virtuelle Verfahren**

Eine Variante der virtuellen Verfahren ist die CAD – basierte Programmierung. Hier werden der Programmablauf und die Bahnkurven auf Basis von Konstruktionszeichnungen erstellt. Dazu sind der Roboter mit verwendetem Werkzeug, die zu bearbeitenden Bauteile sowie die gesamte, aus Schutzzäunen und möglicherweise weiteren Robotern bestehende Umgebung, in einem virtuellen Modell abgebildet. Der gesamte Bewegungsablauf des Roboters wird mit möglichst hohem Automatisierungsgrad am Rechner erstellt. Dazu gehören das Erstellen der Bahnkurve in geeigneter Weise, das Ein- und Ausschalten der Werkzeuge sowie die Festlegung der optimalen Bewegungsgeschwindigkeit.

Eine andere Möglichkeit ist der Einsatz von Funktionen der Augmented Reality. Bei dieser Methode wird mit einem Bildschirm oder einer speziellen Brille das reale Bild um zusätzliche virtuelle Informationen ergänzt. Ein bekanntes Beispiel dieser Technologie ist die virtuelle Hilfslinie zur Kontrolle von Abseitsstellungen beim Fußball. Bei der Roboterprogrammierung kann der Roboter zur Kontrolle der Bewegung als Modell in eine real existierende Zelle einblendet werden. Ohne eine aufwendige Modellierung der kompletten Zelle lassen sich die Reichweiten kontrollieren oder neue Werkzeuge überprüfen. Auch die Einblendung von nützlichen Zusatzinformationen, beispielsweise Koordinatensystemen, ist möglich. Dies ist besonders zu Schulungszwecken sinnvoll. Die Augmented Reality ist ein vielversprechendes und rasant wachsendes Forschungsgebiet innerhalb der virtuellen Realität. Weiterführende Informationen sind in den Quellen [Zäh-05], [Bis-07] und [Vog-08] zu finden.

Ein großer Vorteil der virtuellen Verfahren, insbesondere bei großen Produktionsstraßen mit einer Vielzahl von Robotern, ist die Absicherung grundlegender Fragestellungen. Konkret geht es um die Erreichbarkeit von Arbeitspunkten, die Vermeidung von Kollisionen sowie das schnelle Überprüfen verschiedener Ablaufvarianten oder Robotertypen. Planungsänderungen am Rechner sind wesentlich kostengünstiger durchführbar als nachträgliches Umbauen in der Produktionsstraße. Nachteile ergeben sich bei fehlender Übereinstimmung der realen Umgebung mit dem virtuellen Modell. Versorgungsleitungen für die Roboterwerkzeuge bedingen oftmals erhebliche Bewegungseinschränkungen und sind in der Simulation nicht originalgetreu abzubilden.

In der Praxis werden virtuell erstellte Roboterprogramme auf den Controller übertragen und anschließend an die reale Umgebung angepasst. Der Aufwand für die Nachbearbeitung ist dabei stark von der Modellqualität abhängig. Da in der Regel nur bei größeren Projekten ein virtuelles Modell der Roboterzelle samt Umgebung vorliegt, ist diese Methode für kleine Projekte, dazu zählen die Komponentenversuche in der Fahrzeugentwicklung, nicht geeignet [Web-09].

#### **3.3.4.3 Hybride Verfahren**

Hybride Verfahren zur Programmierung von Industrierobotern verbinden die Vorteile der Online- und Offline – Verfahren miteinander. Auf Basis des geplanten Bewegungsablaufs wird in einem komfortablen Editor mittels Schleifen, Verzweigungen und Bewegungsbefehlen die Grundstruktur des Programms erzeugt. Der Programmierer muss dabei gestützt auf Erfahrungen die Bewegung möglichst genau vor seinen Augen haben und alle denkbaren Be-

sonderheiten berücksichtigen [Web-09]. Nach einer Übertragung dieser ersten Programmversion auf den Robotercontroller folgt im zweiten Schritt die Zuordnung von realen Positionsdaten zu den anfangs leeren Variablen in den Bewegungsbefehlen. Dazu erfolgt eine schrittweise Betrachtung der einzelnen Programmzeilen und parallel dazu die passende Positionierung des Roboterarms. Im Anschluss wird die Position eingelesen und in einer Datenbank auf dem Controller gespeichert. Kommen dabei kleinste Programmierfehler zum Vorschein, lassen sich diese relativ leicht über die Eingabemöglichkeiten am Handbediengerät korrigieren. Sind umfangreiche Änderungen erforderlich, empfiehlt sich erst eine Übertragung auf den Rechner und nach der Bearbeitung wieder zurück zum Robotercontroller.

In der heutigen Praxis eingesetzte Programmierverfahren sind in der Regel hybride Verfahren, die in Abhängigkeit zur Aufgabenstellung aus den beschriebenen Möglichkeiten zusammengesetzt sind. [Bre-10a] stellt ein Programmierverfahren vor, bei dem der Bediener je nach aktueller Situation zwischen dem Online- und dem Offline – Modell der Roboterzelle wechseln kann. Dieses ist ohne Datenverlust möglich, da bei Änderungen im aktiven Modell auch das nicht aktive Modell aktualisiert wird. Eine wesentliche Aufgabe der CAD – basierten Offline – Programmierung, die Erstellung eines Zellenmodells, wird dabei durch Sensorik unterstützt. [Bru-94] beschreibt, wie durch den Einsatz zusätzlicher Sensorik die online auszuführenden Korrekturen der Armpositionierung bei offline erstellten Roboterprogrammen automatisiert werden können und damit der Programmieraufwand zu minimieren ist.

#### 3.3.4.4 Aufgabenorientierte Programmierung

Industrieroboter sind ein universell einsetzbares Hilfsmittel, dessen Komplexität und Funktionsumfang stetig ansteigt. Roboter können daher durch Laien, die keine mehrtägige Programmierschulung besucht haben und über keinerlei Erfahrung verfügen, mit den herstellerseitig verfügbaren Mitteln nicht programmiert werden. Erschwerend kommt hinzu, dass jeder Roboterhersteller eine eigene Philosophie verfolgt und somit keine einheitlichen Standards für eine Programmierungsumgebung vorliegen. Auch der in [Her-09b] beschriebene Ansatz von Microsoft, mit dem Microsoft Robotics Studio ein heterogenes Softwaretool zur Programmierung von Robotern zu etablieren, ist aufgrund von fehlenden Hardwaretreibern bislang nicht erfolgreich.

Ein wichtiger Aspekt zur Förderung der Verbreitung von Robotern und somit der Inanspruchnahme ihrer Vorteile ist eine Vereinfachung der Programmierung. Bei der aufgabenorientierten oder auch impliziten Programmierung wird daher zwischen dem Bediener und der herstellerspezifischen Roboterprogrammiersprache eine zusätzliche Abstraktionsschicht installiert. Diese besteht in der Regel aus relativ wenigen, einfach verständlichen Befehlen sowie einer grafischen Bedienoberfläche und ist für das spätere Einsatzgebiet, beispielsweise Handhabungs-, Mess- oder Schweißanwendungen, optimiert. Die Befehle werden als aufgabenorientiert bezeichnet, da sie keinen direkten Bewegungsbefehl für den Roboter beinhalten. Sie bezeichnen eher konkret, was der Roboter zu tun hat. Aus dem expliziten Befehl *Fahre zu Position A* wird die implizite Forderung *Drücke den Knopf*. Auch zusätzliche Sensorik kann über die aufgabenorientierten Befehle wesentlich verständlicher eingebunden werden. Die Anfänge der impliziten Programmierung liegen in der Weltraumtechnik. Dabei waren besonders die langen Signallaufzeiten von mehreren Sekunden eine große Herausforderung. Im Jahr 1990 wurde an der *Technischen Universität Dortmund* das System *OSIRIS* entwickelt. Auch hier sollte sich die Programmierarbeit der Anwendung mehr auf das *was* als auf das *wie* konzentrieren. Dazu erfolgte eine Aufteilung der Programmiertätigkeiten auf zwei Personen. Ein robotertechnisch hochqualifizierter Softwareentwickler bereitete die Schnittstelle und Programmierungsumgebung soweit vor, dass später ein robotertechnisch unerfahrener Anwender den

Bewegungsablauf einstellen konnte [Bru-01], [Fre-90]. Weitere Informationen zur aufgabenorientierten Programmierung von Industrierobotern sind in den Quellen [Zab-93], [Hum-98], [Kug-99], [Bre-05], [Fin-05], [Pir-05], [Vog-08] und [Web-09] nachzulesen.

### 3.3.5 Förderung der Einsatzmöglichkeiten für Industrieroboter

Im Vergleich zu den Anschaffungskosten für eine Spezialmaschine sind die Kosten für flexibel einsetzbare Industrieroboter wesentlich geringer. Jedoch sind bei Robotern höhere finanzielle Mittel für die Programmierung und Inbetriebnahme anzusetzen. Besonders für kleine und mittelständische Unternehmen stellt dieser Kostenpunkt ein schwer kalkulierbares Risiko dar. Erfahrene Roboterprogrammierer zählen in derartigen Firmen nur selten zum Stammpersonal und die Fertigungsstückzahlen sind vergleichsweise gering [Wul-09]. Laut [Fra-10] lassen sich auf dem Einsatzgebiet von Industrierobotern neue Tendenzen erkennen. Hierzu gehören Produktionssysteme, die sich durch Varianten- und Kapazitätsflexibilität, schnelle Anpassungsfähigkeit bei Produktänderungen und einen variablen Automatisierungsgrad auszeichnen. Zur Sicherung der Wettbewerbsfähigkeit von produzierenden Unternehmen unter diesen und zukünftigen Bedingungen gibt es eine Vielzahl von Ansätzen. Deren Erarbeitung begann in den 1990er Jahren durch Projekte zur Optimierung von Schweißanwendungen. Die Dissertationen [Lie-91], [Zab-93], [Hol-95] und [Pep-97] beschreiben unter anderem Konzepte, wie Schweißfacharbeiter über aufgabenorientierte Makros von der Roboterprogrammierung zu entlasten sind. Dabei erfolgt die automatisierte Ausführung von häufig in Reihenfolge verwendeten Befehlen im Zusammenhang mit grafischen Bedienoberflächen.

Weitere konkrete Projekte, die sich mit der Förderung der Einsatzmöglichkeiten von Industrierobotern in kleinen und mittelständischen Unternehmen auseinandersetzen, sind nachfolgend aufgeführt.

#### **Dissertation Dammertz (1996)**

In der Arbeit *Ein Programmiersystem zur grafisch strukturierten Erstellung von Roboterprogrammen und Programmieroberflächen* wird eine grafische Bedienoberfläche zur Erzeugung von Roboterprogrammen vorgestellt.

Dammertz unterteilt die potentiellen Anwender anhand ihrer Qualifikationen in Roboterexperten, Applikationsspezialisten, einfache Programmierer und Endanwender. Je nach Gruppenzugehörigkeit werden verschiedene Anforderungen an das Programmiersystem definiert. Dem Experten stehen sämtliche Funktionalitäten einer Roboterprogrammiersprache zur Verfügung, Endanwender können lediglich kleine Änderungen an bereits vordefinierten Programmen über eine extra zugeschnittene Bedienoberfläche vornehmen.

Diese Unterteilung war erforderlich, da aufgrund der zunehmenden Vernetzung von Robotern und Integration von zusätzlichen Sensoren der Anteil an allgemein leicht verständlichen Bewegungsbefehlen im Programmcode immer weiter sinkt, teilweise auf unter zehn Prozent. Dafür steigt der Anteil von komplexen Strukturen, Kommunikationsbefehlen und der Sensordatenverarbeitung, was ein einheitliches Programmiersystem für alle Bediener verhindert [Dam-96].

**Dissertation Lüdemann – Ravit (2005)**

Die Dissertation *Ein System zur Automatisierung der Planung und Programmierung von industriellen Roboterapplikationen* schafft eine herstellerunabhängige und steuerungsneutrale Programmierungsumgebung.

Deren Notwendigkeit ergibt sich daraus, dass die Entwicklung einer einheitlichen Roboterprogrammiersprache gescheitert ist. Das erstellte Programm kann zur späteren Ausführung auf *jede beliebige* Steuerung übertragen werden.

Ein in das Gesamtsystem integriertes Softwaretool ist in der Lage, verschiedene Varianten zur Bewältigung einer Roboteranwendung zu simulieren und zu bewerten. Die Varianten werden dabei anhand eines sogenannten Generierungsplans automatisch erzeugt, dieser ist durch einen erfahrenen Roboterprogrammierer erstellt worden. Vorhandene und geprüfte Generierungspläne für eine konkrete Aufgabe helfen später unerfahrenen Anwendern bei der Bearbeitung von vergleichbaren Aufgabenstellungen [Lüd-05].

**Forschungsprojekt MORPHA (1999 – 2003)**

Das Projekt *MORPHA* wurde unter der Beteiligung von unterschiedlich großen Forschungseinrichtungen und Betrieben aus dem industriellen Umfeld durchgeführt. Das Hauptziel lag in der Ausstattung von Robotern mit leistungsfähigen Kommunikations-, Interaktions- und Verhaltensfähigkeiten.

Ein Teilprojekt dieses Vorhabens befasste sich mit der Methode *Programmieren durch Vormachen*. Eine vom Menschen ausgeführte Handlung wird von einer Videokamera sowie zusätzlicher Sensorik aufgenommen und anschließend in verschiedene Teilhandlungen segmentiert. Aus den jeweils gefundenen Segmenten erfolgt die Beschreibung von Handlungen, beispielsweise Greifen, Bewegen oder Loslassen. In den nächsten Schritten generiert das System den roboterspezifischen Programmcode und simuliert die Bewegungen. Nach einer Freigabe durch den Bediener wiederholt der Roboter die vom Menschen vorgemachte Aktion.

Praktische Tests mit einem mobilen Robotersystem verliefen erfolgreich, der Roboter deckte dabei einen Tisch mit Geschirr. Das langfristige Ziel ist, durch neue Programmier- und Bedienkonsolen den Weg für leistungsfähige Assistenzsysteme in den Bereichen Produktion, Haushalt und Pflege zu bereiten. Weitere Informationen sind in den Quellen [Dil-00], [Ehr-03] und [Bre-09a] zu finden.

**Forschungsprojekt A – Bot (2006 – 2011)**

Das universitäre Forschungsprojekt *A – Bot* basiert auf dem Ansatz, die Programmierung von Handhabungsprozessen durch den Einsatz von externer Sensorik zu vereinfachen.

Toleranzbedingte Abweichungen bei zu greifenden Objekten werden durch Kamerasysteme sowie Kraft- oder Momentsensoren erfasst und bewirken eine Korrektur der zuvor eingelernten Positionen. Zur Vereinfachung der Programmierung ist die Datenverarbeitung der externen Sensorik direkt in den Robotercontroller integriert. Andernfalls müsste der Programmierer über tiefgreifende Kenntnisse auf dem Gebiet der Messdatenverarbeitung sowie der Einbindung von Sensoransteuerungen in den Programmcode verfügen.



Für das im Rahmen einer Dissertation erarbeitete Konzept wird der industrielle Einsatz angestrebt [Hen-09].

### **Unscharfe Roboterprogrammierung (2000 – 2007)**

Die Methode *Unscharfe Roboterprogrammierung*, die in ähnlicher Form ebenso im Ansatz *Automatic Path Learning* entwickelt wurde, bewirkt eine Verkürzung der Inbetriebnahme und optimiert die Bahnkurven.

Grundsätzlich wird die auszuführende Bewegung nur relativ grob durch den Programmierer vorgegeben, dies kann beispielsweise durch Führen des Roboters mit der Hand oder auf Basis virtueller Modelle geschehen. Die genaue Positionierung während der Bewegungsausführung wird mit Hilfe eines hochauflösenden Sensors berechnet.

Diese Methode eignet sich für Bearbeitungsprozesse, da diese bedingt durch Fertigungstoleranzen oder unvorhersehbare Widerstände sicherheitshalber mit sehr langsamen Geschwindigkeiten ausgeführt werden. Zudem wird die Programmierung einer Handhabungsaufgabe mit ungenau positionierten Werkstücken erleichtert. Weitere Informationen sind in den Quellen [Wec-06] und [Her-08] zu finden.

### **Forschungsprojekt *IRoProg* (2003 – 2005)**

Das Forschungsprojekt *IRoProg* hatte zum Ziel, den Programmieraufwand von Industrierobotern von der ersten Offline – Programmierung bis zur letzten Online – Optimierung deutlich zu reduzieren. Dadurch soll sich der Einsatz von Robotern auch bei kleinen Losgrößen rentieren.

Zum Zweck der Optimierung offline erstellter Programme kommen ein roboterähnlicher Manipulator, eine optisch verbesserte virtuelle Zellendarstellung sowie optimierte Verfahren zur kollisionsfreien Bahnplanung zum Einsatz. Dadurch steigt die Qualität von offline erstellten Bahnkurven. Zusammen mit neuen Methoden der Kalibrierung kann im Anschluss die Inbetriebnahme im Online – Modus schneller erfolgen. Innovative Ansätze bei der Onlineprogrammierung, beispielsweise ein einfaches *Zeigen mit dem Finger* auf die Zielposition, sollen das manuelle Bewegen des Roboterarms und einfache Handlungen beim Nachteachen auch ungeschulten Mitarbeitern ermöglichen.

Die Erprobung der entwickelten Ansätze im industriellen Einsatz war erfolgreich und ist über die Bearbeitungsdauer hinaus im Einsatz [Den-05].

### **Forschungsprojekt *PORTHOS* (2003 – 2006)**

Am Verbundprojekt *PORTHOS* waren Forschungseinrichtungen, Hersteller von Roboter- und Sicherheitstechnik sowie Systemlieferanten beteiligt. Der Schwerpunkt lag in der Entwicklung eines kurzfristig einsatzfähigen Gesamtsystems, welches flexibel zur Erweiterung von Produktionsanlagen einsetzbar ist. Darüber hinaus muss eine Rentabilität bei kleinen Losgrößen und häufig wechselnden Einsatzorten gegeben sein.

Die Lösung des Problems ist eine leichte Transportier- und Adaptierbarkeit der Hardware, eine integrierte Absicherung des Gefahrenbereichs sowie eine Bedienbarkeit durch nicht spezialisiertes Personal. Der Schwerpunkt des Projekts liegt in der Anpassungsfähigkeit des Gesamtsystems an wechselnde Einsatzbedingungen. Erreichbar wird diese durch den Einsatz von Sensoren zur Unterstützung der Kalibrierung, der Korrektur von Roboterbewegungen und der Greiftechnik.

Die Validierung der Entwicklung fand im praxisnahen Einsatz bei der Bestückung einer Beschneidpresse mit Töpfen statt. Weiterführende Informationen sind den Quellen [Ahl-04], [Bre-04], [Bre-05] und [Mat-06] zu entnehmen.

#### **Forschungsprojekt *Rent – A – Robot* (2005 - 2008)**

Unter Mitwirkung von acht Projektpartnern, bestehend aus Forschungseinrichtungen, Systemintegratoren und Endanwendern, leistet das Projekt *Rent – A – Robot* einen Beitrag zur Erhaltung von wirtschaftlich ausführbaren Montagetätigkeiten.

Das Projekt gliedert sich in zwei Bereiche. Zum einen wird ein technisches Konzept vorgestellt, welches den temporären Einsatz von Industrierobotern in Produktionsanlagen ermöglicht. Wesentliche Bestandteile sind die Mobilität der Anlage, eine aufgabenorientierte und herstellerunabhängige Programmierung, die einfache hard- und softwaretechnische Adaptierbarkeit an bestehende Einrichtungen sowie die Sicherheitstechnik. Der zweite wirtschaftliche Projektteil erörtert das Geschäftsmodell einer Betreiber- und Servicegesellschaft, welche die Beschaffung, Vermietung und Inbetriebnahme der technischen Anlage in Form von Betreibermodellen beim potentiellen Kunden vorsieht.

Die Erprobung eines Prototyps erfolgte bei zwei Projektteilnehmern im praxisnahen Umfeld. Weitere Information sind in den Quellen [Bre-08a] und [Bre-08b] zu finden.

#### **Forschungsprojekt *ProDemo* (2007 – 2009)**

Das Ziel des Forschungsprojekts *ProDemo* ist, die Grenze zwischen Online- und Offline – Programmierverfahren aufzuheben. Dadurch soll die Hürde für den Robotereinsatz in kleinen und mittelständischen Unternehmen gesenkt werden, da Inbetriebnahmen zeitoptimiert und leichter verständlich ausgeführt werden können.

Ein Ansatz verfolgt die Optimierung der aufwendige Offline – Modellierung einer Roboterzelle durch ein Online – Zeigegerät. Mit diesem Zeigegerät und einer Kamera lassen sich Konturen von Hindernissen im Arbeitsbereich des Roboters unverzüglich in das Zellenmodell einfügen. Die Kamera zeichnet zudem die Bahnkurve des vom Bediener mit der Hand geführten Werkstücks auf. Mit Hilfe dieser Aufzeichnungen und einem Algorithmus zur Werkstückverfolgung lassen sich Bahnkurven ohne zeitaufwendige Positioniervorgänge des Roboterarms im Online – Modus generieren [Aut-09], [Bre-09b].

#### **Forschungsprojekt *SMERobot* (2005 – 2009)**

*SMERobot* war eine vielschichtige europäische Initiative mit einer großen Anzahl an teilnehmenden Forschungseinrichtungen, Roboterherstellern, Systemlieferanten und Endanwendern. Das entwickelte Systemkonzept soll in den unterschiedlichsten Bereichen die Inbetriebnahme

von Robotern in maximal drei Tagen gestatten und eine Programmierbarkeit *durch Jedermann* ermöglichen.

Dazu dienen verschiedene parallel erarbeitete Methoden, zum Beispiel das sogenannte *Vormachen*, die Programmierung via Spracheingabe oder ein stiftähnlicher Sensor als Teachsystern auf Basis von Bewegungssensoren. Zur Ausführung von verschiedenen Handhabungsaufgaben dienen mit Sensorik ausgestattete Greifwerkzeuge. Ein weiterer Teil des Projekts bestand in der Verwirklichung einer Kooperation von Mensch und Roboter, beispielsweise zur Unterstützung bei körperlich schweren Schweißarbeiten. Dazu wurde ein Sicherheitskonzept für die direkte Mensch – Roboter – Kooperation erstellt. Die Vision sieht Roboter zukünftig reibungslos als *ganz normale Werkzeuge* in den Werkstattalltag integriert.

Die Erprobung der Konzepte fand in vielen Einsatzbereichen, wie Holzverarbeitung, Schweißtechnik, Gießereitechnik oder Anlagenbestückung im industriellen Umfeld statt. Das Projekt ist in den Quellen [Hol-06], [Häg-09] und [Wes-09] detailliert aufgeführt.

### **Softwareschnittstelle *Robotics Library***

Die von der Firma *ImagingLab* entwickelte Softwareschnittstelle beinhaltet Funktionen, welche eine Ansteuerung von Industrierobotern der Hersteller *KUKA*, *Denso* und *Mitsubishi* über LabVIEW ermöglichen. Wird die allgemeine Programmierung der Roboter auch nicht entscheidend vereinfacht, besteht jedoch die Möglichkeit der Verknüpfung von Robotik und zusätzlicher Messtechnik. Programmierer, die über Erfahrungen in LabVIEW verfügen, müssen keine neue Programmiersprache erlernen, lediglich allgemeines Grundlagenwissen über die Funktions- und Arbeitsweise von Robotern wird vorausgesetzt. Die Schnittstelle leistet einen Beitrag zur Vereinheitlichung der Programmierung von Industrierobotern. Weitere Informationen können der Quelle [Ima-10] entnommen werden.

### **3.3.6 Mensch – Roboter – Kooperation**

Die beschriebenen vielfältigen Möglichkeiten zur Ausweitung des Einsatzes von Industrierobotern und die damit verbundenen Einsparpotentiale haben dazu geführt, dass heutzutage Menschen mit unmittelbarer Unterstützung durch Roboter Arbeiten ausführen. Die Vorteile der menschlichen Arbeitskraft vereinen sich mit denen einer Maschine. Menschen verfügen über Wissen, Geschicklichkeit, Entscheidungsfähigkeit sowie eine sehr feine Sensorik. Roboter zeichnen sich durch Ausdauer, Genauigkeit, Schnelligkeit und Kraft aus [Schr-03]. Aufgaben lassen sich nur dann gemeinsam ausführen, wenn Mensch und Roboter nicht durch Schutzeinrichtungen voneinander getrennt sind. Es besteht daher eine Reihe von Ansätzen und Lösungen, um die Sicherheit des Menschen auch ohne Schutzzäune gewährleisten zu können.

Der Verfasser von [Mau-07] beschreibt eine Möglichkeit, durch ein Multikamerasystem bewegliche Objekte im Arbeitsraum des Roboters zu identifizieren. Je nach Abstand des Objekts, zum Beispiel eines Menschen, zum Roboter, wird die Bewegungsgeschwindigkeit stufenlos variiert. Die Geschwindigkeit kann dabei im Millisekundentakt soweit verringert werden, dass der Roboterarm nicht mehr als Verletzungsquelle gilt. [Hue-10] definiert in einer Liste Maximalkräfte, die im Fall einer Kollision an den verschiedenen menschlichen Körperteilen einwirken dürfen. Auf dieser Basis besteht die Möglichkeit, verletzungsarme Oberflächen am Roboterarm zu montieren. [Wör-03] stellt dazu eine taktile Schutzhaut vor, die zur Sicherheit des Menschen Kontaktkräfte am Roboterarm erkennt und die Bewegung stoppt.

Ein zusätzlicher Anreiz zur Entwicklung neuer Sicherheitskonzepte sind finanzielle Gründe. Aufwendungen für Schutzeinrichtungen übersteigen die Kosten eines Roboters bei weitem.

### **3.3.7 Zusammenfassung**

Industrieroboter sind hoch flexible und vergleichsweise kostengünstige Maschinen zur Erledigung einer stetig steigenden Fülle potentieller Aufgaben. Dieses wird durch den Einsatz von optionaler Sensorik zusätzlich gefördert. Mit der rapiden Zunahme der Verbreitung von Robotern und den dadurch in der Summe gestiegenen Kundenanforderungen ist deren Programmierung komplizierter geworden. Zudem besteht keine einheitliche Programmiersprache. Zur Inbetriebnahme von Industrierobotern bedarf es grundlegender Kenntnisse über den technischen Aufbau, die Bewegung des Arms sowie über hersteller- und typenspezifische Eigenheiten der Programmierung. Je höher der Einrichtungsaufwand im Vergleich zur eigentlichen Anwendungsdauer ist, desto geringer wird der wirtschaftliche Nutzen.

Betriebe mit großen Produktionsstraßen, hunderten von Robotern und teilweise mehrjährigen Produktionszyklen passen sich an diese Rahmenbedingungen an. Kleine und mittelständische Unternehmen mit geringeren Losgrößen und weniger geschulten Roboterprogrammierern benötigen zusätzliche Hilfsmittel, um das Verlustrisiko zu minimieren. Ein Hilfsmittel ist die aufgabenorientierte Programmierung. Je nach Anwendungsgebiet, beispielsweise als Schweißroboter, Handhabungsmaschine oder in der Messtechnik, bedarf es einer in enger Zusammenarbeit mit dem Bediener entwickelten Schnittstelle. So müssen die im Vorfeld definierten Arbeitsabläufe durch die Bediener ohne Roboterschulung intuitiv und schnell in Betrieb genommen werden können. Der dafür verfügbare Funktionsumfang muss das optimale Mittelmaß zwischen Flexibilität und Verständlichkeit aufweisen. Nur dann ist das häufig formulierte Ziel, Roboter binnen weniger Minuten zu programmieren, erreichbar.

Für die Erstellung und Anpassung dieser aufgabenorientierten Schnittstelle sind Roboterexperten verantwortlich, die mehrere Roboterzellen betreuen und im engen Kontakt zu den Bedienern stehen. Strukturelle Änderungen am Programm können auch bei zukünftigen Änderungen nur durch den Roboterexperten durchgeführt werden. Gelingt die Einführung und Anwendung dieser aufgabenorientierten Schnittstelle in einem Anwendungsbereich, so sind durch Industrieroboter große Einsparungen möglich. Scheitert die Umsetzung an technischen Problemen oder an der Akzeptanz der Bediener, wird sich der Einsatz von Robotern in diesem Bereich kaum rentieren.

## **3.4 Fahrzeugelektronik**

Die wesentlichen technischen Anforderungen des Kunden an eine neue Fahrzeuggeneration sind neben einer hohen Qualität ein Zugewinn an Komfort, weniger Kraftstoffverbrauch und geringere Schadstoffemissionen sowie Weiterentwicklungen für mehr Sicherheit. Zur Erfüllung dieser Kriterien ist der Anteil elektronischer Komponenten im Fahrzeug mit jeder neuen Generation angestiegen. Anfangs bestanden diese Komponenten nur aus elektrischen Bauteilen, wie Lampen, Kabeln und Schaltern. Durch den Einsatz von Steuergeräten und darin verbauten Mikroprozessoren hielt auch die Software Einzug ins Automobil.

### 3.4.1 Steuergeräte

In modernen Kraftfahrzeugen ist heute eine Vielzahl von Steuergeräten für die verschiedensten Aufgaben verbaut. Beispiele sind Motor-, Airbag- oder Türsteuergeräte. Die örtliche Verteilung dieser Steuergeräte im Fahrzeug ist so konzipiert, dass zur Verbindung der angeschlossenen Sensoren und Aktoren möglichst kurze Leitungslängen zur Optimierung der Fahrzeugmasse nötig sind. Die eingelesenen Sensordaten werden nach dem Prinzip *Eingabe – Verarbeitung – Ausgabe (EVA)* in Ausgangssignale für die Aktoren umgewandelt. Diese Umwandlung erfolgt durch mathematische Algorithmen unter der Berücksichtigung von im Steuergerät hinterlegten Parametern und Kennlinien. Letztgenannte können beispielsweise bei verschiedenen Motortypen variieren oder durch das Drücken der *Sport – Taste* verändert werden [Rei-08].

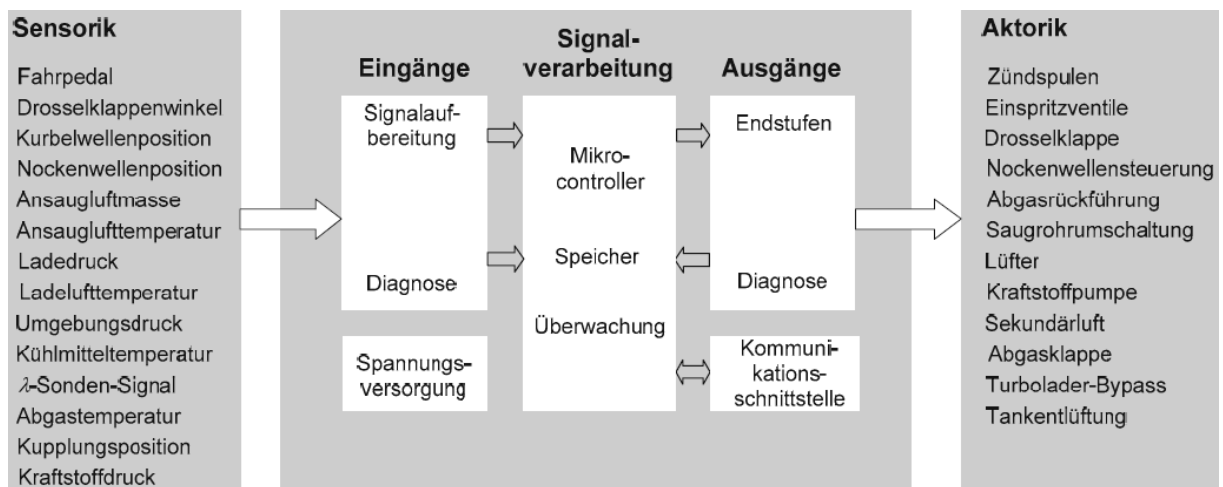


Abbildung 18: Aufbau eines Steuergeräts [Rei-08]

Zur Reduzierung von Kosten durch Redundanzen in der Sensorik oder Berechnung sind sämtliche Steuergeräte im Fahrzeug über Bussysteme zu einem Netzwerk verbunden. Dadurch stehen dem einzelnen Steuergerät nicht nur die eigenen Sensordaten, sondern prinzipiell alle im Fahrzeug vorliegenden Daten zur Verfügung.

### 3.4.2 Bussysteme

Zur Vernetzung der Steuergeräte werden in Abhängigkeit von der benötigten Datenrate sowie den Sicherheits- und Echtzeitanforderungen die Bussysteme LIN, CAN, FlexRay oder MOST eingesetzt.

Der LIN – Bus ist eine kostengünstige und mit 20 Kilobit pro Sekunde vergleichsweise langsame Vernetzungsart mit nur geringen Fehlererkennungsmechanismen. Der Einsatz ist für die Verbindung von Steuergeräten mit sogenannten intelligenten Sensoren und Aktoren, also solchen mit eigenem Mikroprozessor, konzipiert. Ein Beispiel ist der Schalterblock in der Fahrertür zur Bedienung der Fensterheber und Außenspiegel. Ohne einen Datenbus wäre für die Übertragung des Status der Schalter eine Vielzahl von Leitungen nötig. Durch den LIN – Bus reicht eine einzige Datenleitung aus.

Der CAN – Bus wurde im Jahr 1987 vorgestellt und war die erste Technologie zur Vernetzung von Steuergeräten. Heute existieren mit der Low – Speed (100 Kilobit pro Sekunde) und High – Speed (500 Kilobit pro Sekunde) zwei Geschwindigkeitsvarianten mit gleichen Me-

chanismen, die eine fehlerfreie Datenübertragung im Netzwerk sicherstellen. Die Entscheidung für die optimale Variante wird in Abhängigkeit von Leitungslänge, Busauslastung und benötigter Übertragungsgeschwindigkeit getroffen. Über den CAN – Bus werden in erster Linie Sensorsignale und Berechnungsergebnisse durch das sendende Steuergerät allen Busteilnehmern zur Verfügung gestellt. Steuergeräte auf der Empfangsseite verarbeiten nur die benötigten Signale, die übrigen Daten auf dem CAN – Bus werden ignoriert.

Mit Blick auf die zukünftigen Anforderungen an die Vernetzung im Fahrzeug wurde FlexRay entwickelt. Die maximale Übertragungsrate dieses Bussystems beträgt 10 Megabit pro Sekunde, es ist echtzeitfähig und verfügt über eine hohe Fehlersicherheit. Bei zukünftigen Brake – by – Wire oder Steer – by – Wire – Anwendungen sind diese Eigenschaften bedingt durch die fehlende direkte mechanische Verbindung zwingend notwendig. Physikalisch verfügt FlexRay über zwei Kanäle, die bei sicherheitsrelevanten Nachrichten zur redundanten Übertragung oder ansonsten zur Verdoppelung der Bandbreite nutzbar sind. In der Praxis werden analog zum CAN – Bus Mess-, Steuer- und Reglersignale übertragen.

Der MOST – Bus wurde konzipiert, um Daten zwischen Multimediageräten, beispielsweise einem Navigationssystem, Autoradio oder DVD – Wechsler, auszutauschen. Diese Anwendungen haben keine besonderen Anforderungen an Echtzeitfähigkeit oder Sicherheit. Die benötigte Datenrate übersteigt jedoch deutlich die von Messdaten und Regelsignalen. Bei MOST150 liegt die Datenübertragungsrate bei 150 Megabit pro Sekunde.

Der Einzug und die stetige Zunahme von elektronischen Funktionen im Kraftfahrzeug erhöhen die potentiellen Fehlerquellen enorm. Damit diese Systeme beherrschbar sind und im Fall von Störungen eine Lokalisierung der Ursache möglich ist, bedarf es zusätzlicher Überwachungs- und Diagnosemöglichkeiten [Rei-08], [Pet-08], [Zim-10].

### **3.4.3 On – Board – Diagnose**

Als On – Board – Diagnose wird die Eigenschaft eines Steuergeräts bezeichnet, sich innerhalb des gesamten Lebenszyklus permanent selbst zu überwachen. Diese Überwachung beginnt bei der ersten Versorgung mit der benötigten Betriebsspannung während der Produktion und endet mit der Stilllegung des Fahrzeugs. Zur Diagnose zählen die Kontrolle elektrischer Bauteile auf Kurzschlüsse oder Kabelbrüche, Plausibilitätsprüfungen oder Vergleiche von Messergebnissen sowie eine Überwachung der Aktorik. Ferner zählen eine Konsistenzprüfung der über Bussysteme empfangenen Daten sowie eine Prüfung der im Steuergerät gespeicherten Parameter und Kennlinien dazu. Für Komponenten und Funktionen, die eine Auswirkung auf Emissionswerte haben, sind vom Gesetzgeber ununterbrochene Kontrollen durch die On – Board – Diagnose vorgeschrieben.

Treten während dieser automatischen Überwachung Fehler auf, folgt ein Eintrag in den nicht-flüchtigen Fehlerspeicher des Steuergeräts. Soweit möglich, werden diesem Fehlerspeichereintrag sinnvolle Zusatzinformationen, beispielsweise Außentemperatur, Bordnetzspannung oder Fahrzeuggeschwindigkeit zum Fehlerzeitpunkt, hinzugefügt. Ist der Fehler schwerwiegend, wie beispielsweise der Ausfall einer Zündkerze, kann dies zu Beschädigungen des Fahrzeugs führen und wird durch das Aufleuchten einer Kontrolllampe im Kombiinstrument signalisiert. Bei einem außerplanmäßigen, oder im weniger kritischen Fall beim nächsten planmäßigen Werkstattaufenthalt, werden die Fehlercodes in den Fehlerspeichern aller Steuergeräte durch ein Diagnosegerät ausgelesen und analysiert. Zum Auslesen der Daten ist dazu ein vom Fahrersitz aus erreichbarer genormter Stecker gesetzlich vorgeschrieben. Der Mechaniker bekommt im Optimalfall anhand der Kundenbeschreibung und der Fehlercodes eine

genaue Vorgehensweise vom Diagnosegerät angezeigt, wie das vorhandene Problem zu beheben ist. Eine aufwendige und technisch sehr anspruchsvolle manuelle Fehlersuche, die nur mit umfangreichem Wissen über das Gesamtsystem möglich ist, soll so entfallen. Sind für das Fahrzeug seit dem letzten Werkstattaufenthalt neuere Softwareversionen in der Form von Programmcodes, Parametern oder Kennlinien vorhanden, können diese mit dem Diagnosegerät auf das Steuergerät übertragen werden. Dieser Vorgang wird *Flashen* genannt.

Auch während der Entwicklung und Absicherung von Fahrzeugen spielt die Diagnose – Funktion eine wichtige Rolle. Durch diese erhält man Zugriff auf die internen Zustände eines Steuergeräts, die nicht über die Bussysteme nach außen kommuniziert werden. Die Ausführung von Diagnosebefehlen und das Auslesen des Fehlerspeichers während der Produktion dienen dazu, defekte Bauteile oder Montagefehler direkt zu erkennen und zu beheben. Dabei ist zwischen *echten Fehlern* und *Scheinfehlern* zu unterscheiden. Scheinfehler treten auf, wenn der Mikroprozessor im Steuergerät bereits arbeitet, die für einen reibungslosen Betrieb benötigten Hardwarekomponenten aber noch nicht angeschlossen sind [Wal-06], [Pet-08], [Rei-08], [Zim-10].

### 3.4.3.1 ODX

Die Speicherung von Daten im Steuergerät, beispielsweise Programmcode, Parameter, Kennlinien oder Fehlerspeichereinträge, sowie die Datenübertragung vom und zum Fahrzeug über die Bussysteme erfolgen informationstechnisch in der Form von Datenbytes. Diese Datenbytes sind auf den ersten Blick nicht verständlich und müssen in eine lesbare Darstellungsform übersetzt werden. Die dafür benötigten Zusatzinformationen sind in einer zum jeweiligen Steuergerät passenden ODX – Beschreibungsdatei hinterlegt. Hauptbestandteil dieser ODX – Beschreibungsdatei sind die sogenannten einfachen Datenobjekte *DOP*. Anhand dieser DOP können über einen Diagnosejob erhaltene verschlüsselte Daten in verständliche Informationen umgewandelt und auf dem Diagnosegerät angezeigt werden.

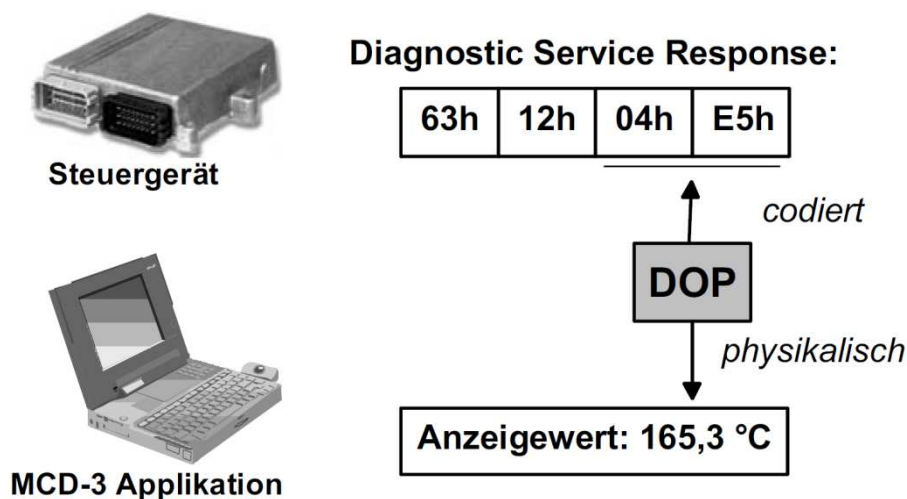


Abbildung 19: Anwendung der DOP zur Anzeige einer Temperatur [Zim-10]

Daneben enthält die ODX – Beschreibungsdatei Informationen, die das Diagnosegerät für das Flashen von Steuergeräten benötigt sowie die zu übertragenden Parameter, Kennlinien und den Programmcode.

Während des Lebenszyklus eines Fahrzeugs sind Änderungen an ODX – Beschreibungsdateien, bedingt durch neue Funktionen, Fehlerbeseitigungen und Optimierungen, unvermeidlich. Die ODX – Dateien müssen daher während der Entwicklung und Produktion zwischen

dem Hersteller, den Zulieferfirmen sowie später den Servicewerkstätten problemlos aktualisierbar sein. Bei Änderungen an einzelnen Elementen in einer ODX – Datei ist es wichtig, dass mittels Versionierung, Änderungsdatum und Dokumentation eine Historie verfügbar ist. Grundsätzlich sollen alle Prozessteilnehmer auf eine gemeinsame ODX – Datenquelle zugreifen und Änderungen ebenfalls nur an dieser Quelle vornehmen. Die Qualität wird somit verbessert, da Fehler für jeden Teilnehmer behoben werden. Weitere Informationen sind in den Quellen [Wal-06], [Rei-08] und [Zim-10] verfügbar.

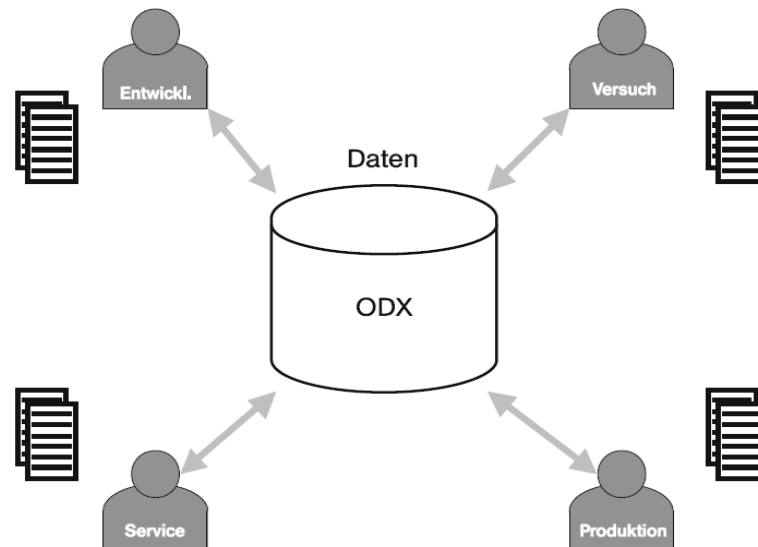


Abbildung 20: Gemeinsamer Zugriff auf ODX – Daten [Wal-06]

#### 3.4.4 Restbussimulation

Die erste Phase der Projektierung eines Fahrzeugbordnetzes ist die Erstellung eines allgemeingültigen Gesamtentwurfs der Architektur. Dieser beinhaltet bis zu 70 Steuergeräte, die über 2.500 Busnachrichten untereinander austauschen. Sämtliche Busnachrichten mit den darin enthaltenen Signalen sind samt Sender- und Empfängersteuergerät, auch Knoten genannt, in einer Kommunikationsmatrix definiert [Bar-11].

Anschließend erfolgt die Entwicklung der Steuergeräte gewöhnlich parallel und individuell in verschiedenen Fachabteilungen und bei Zulieferfirmen. Jede kontinuierliche Ausbau- oder auch Integrationsstufe des Fahrzeugbordnetzes bedeutet eine Zunahme und Optimierung der Funktionalitäten der einzelnen Steuergeräte. Damit steigt die Anzahl der Informationen, die zur fehlerfreien Funktionsweise von den übrigen Knoten des verteilten Netzes benötigt werden.

Zur Absicherung von einzelnen Steuergeräten ist daher eine Nachbildung des Fahrzeugbordnetzes mit sämtlichen darin enthaltenen Knoten durch Modelle notwendig, da ein realer Testaufbau noch nicht zur Verfügung steht. Diese Nachbildung wird als Restbussimulation bezeichnet. Durch die Zunahme der elektronischen Funktionen gewinnt diese Vorgehensweise stetig an Bedeutung, ansonsten wären die Inbetriebnahme und Tests mechanischer Komponenten unabhängig vom Gesamtfahrzeug nicht möglich. Die Aufgaben der Restbussimulation erstrecken sich von einfachen Funktionstests des Zusammenspiels zweier Komponenten im Netzwerk bis hin zur Abbildung des gesamten Fahrzeugbordnetzes. Die generierten Modelle sind in der Lage, zyklisch oder ereignisgesteuert Busnachrichten zu versenden. Sie müssen aber auch inkrementelle Zähler, sogenannte *Alive – Counter*, generieren und Prüfsummen berechnen, die zur Absicherung der Datenübertragung bei kritischen Busnachrichten implemen-



tiert sind. Hinzu kommen realitätsnahe und in Echtzeit veränderliche Messwerte mit Abhängigkeiten von weiteren Signalen. Die Generierung der Netzknoten – Modelle für eine Restbussimulation ist ein weitgehend automatisierter Prozess auf Basis von Datenbankinformationen. *Alive – Counter* und ähnlich komplexe Funktionen können durch Programmiersprachen, die in den Simulationstools integriert sind, implementiert werden [Vec-11a].

Die Tools müssen die Ausführung der simulierten Knoten innerhalb bestimmter zeitlicher Grenzen sicherstellen, damit die von den realen Steuergeräten benötigten Informationen rechtzeitig zur Verfügung stehen. Durch die *harten* Echtzeitanforderungen und die vergleichsweise schnelleren Übertragungsraten von FlexRay sind bei diesem Bussystem die Anforderungen gegenüber den CAN – Restbussimulationen gestiegen. Zur Simulation von FlexRay – Netzwerken sind daher echtzeitfähige Hardwareplattformen notwendig, die eine Versendung der Busnachrichten zu fest definierten Zeitpunkten garantieren.

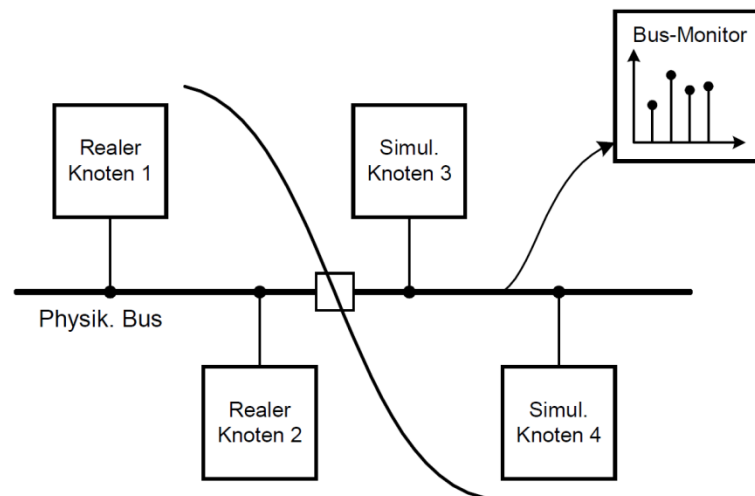


Abbildung 21: Restbussimulation mit zwei realen und zwei simulierten Netzknoten [Göh-11]

In der Integrationsphase der Bordnetzentwicklung erfolgt schrittweise der Austausch von simulierten Knoten durch reale Steuergeräte am Bus. Das Simulationstool übernimmt dann primär die Aufgaben eines Analysewerkzeugs, mit dessen Hilfe über den Bus – Monitor der Datenfluss überwacht und mit den in der Kommunikationsmatrix definierten Anforderungen abgeglichen wird. Die Tools zur Bereitstellung von Restbussimulationen sind ein komplexer und unverzichtbarer Bestandteil bei der Absicherung von Fahrzeugkomponenten. Wichtige Argumente für die Auswahl eines geeigneten Tools sind die Verfügbarkeit sowie die Möglichkeiten zur Manipulation der Knotenmodelle, die Bedien- und Visualisierungsmöglichkeiten sowie die Schnittstellen zur Einbindung in Prüfsystemkonzepte [Hof-06], [Schl-09], [Egg-10], [Göh-11].

## 4 Ein Prüfsystemkonzept für den Komponentenversuch

Die Ausführungen in Kapitel 2 verdeutlichen die Notwendigkeit zur Durchführung von realen Versuchen für die Absicherung und Freigabe von Fahrzeugentwicklungen. Kapitel 3 beschreibt die vorherrschende Situation in den Prüflaboren sowie technisch relevante Grundlagen zur Robotik und Fahrzeugelektronik. Das durch diese Arbeit entwickelte Prüfsystemkonzept beruht auf diesen technischen Gegebenheiten sowie auf vielseitigen Anforderungen, welche zu Beginn dieses Kapitels ausgearbeitet werden. Diese Neuentwicklung ist nötig, da die beschriebenen veränderten Rahmenbedingungen im Automobilbau elementare Verbesserungen der Prüftechnik erfordern. Zwar wurden vorhandene Prüfstände in den vergangenen Jahren im Hinblick auf innovative Technologien erweitert, eine substanzielle Überarbeitung blieb aber aus. Dieser notwendige Schritt wird im Rahmen dieser Arbeit vollzogen.

Kapitel 4 dokumentiert die Entwicklung eines grundlegenden Konzepts zur Automatisierung von Prüfungsaufgaben, welches durch eine hohe Funktionalität und Flexibilität zukunftsorientierte Methoden mit höchstmöglicher Effektivität und Effizienz für die Komponentenerprobung bereitstellt. Inhaltliche Schwerpunkte sind weiterhin die Aufnahme, Verwaltung und Analyse von Messdaten sowie die Einbindung der Robotik. Die Verwendung von Robotern zur Ausführung mechanischer Betätigungen war durch die Ergebnisse der Diplomarbeit [Vog-07a] im Vorfeld festgelegt. Wesentlich ist ebenso die Sicherstellung einer langfristigen Verwendbarkeit der Prüfeinrichtung in der Praxis, auch über den Bearbeitungszeitraum dieses Projekts hinaus.

Die Entwicklungsarbeiten fanden im sehr praxisnahen Umfeld und in ständigem Kontakt mit Fachstellen bei Automobilherstellern und Zulieferfirmen statt. Im Besonderen war auf die Vereinbarkeit der neuen Prüfmethodiken mit bestehenden Prüfvorschriften zu achten. Das Ergebnis ist keine klassische *Einzellösung* für individuelle Anforderungen, es bietet vielmehr im Umfeld der Komponentenerprobung die Basis für eine Vielzahl von Prüfungen, da eine große Schnittmenge mit gängigen Prüfmitteln und Prüfabläufen besteht.

Der Nachweis für eine praxistaugliche Umsetzbarkeit des Prüfsystemkonzepts wird in den Kapiteln 5 und 6 anhand von zwei konkreten Beispielen, einem Prüfstand zur Dauererprobung von Fahrzeugtüren sowie einem Prüfstand für Steifigkeitsmessungen, erbracht. In Kapitel 4 sind zur Veranschaulichung von Sachverhalten vermehrt Beispiele aufgeführt, die sich an Fahrzeugtüren orientieren. Dies soll nicht darüber hinwegtäuschen, dass das Prüfsystem allgemein für Automatisierungsaufgaben in der Komponentenerprobung konzipiert ist.

### 4.1 Anforderungen

Das Klarstellen und Präzisieren einer Aufgabenstellung steht grundsätzlich am Anfang eines Entwicklungsprozesses und ist eine wichtige Grundlage für den späteren Erfolg des Produkts. Als Ergebnis entsteht eine Anforderungsliste, welche übergeordnete Hauptmerkmale definiert und diese in mehreren Abstufungen aufschlüsselt. Die letzte Stufe beschreibt konkret auf das Projekt bezogene Anforderungen [Pah-07].

### 4.1.1 Anforderungsliste

[Pah-07] definiert für eine strukturierte Anforderungsliste eine Leitlinie mit 17 Hauptmerkmalen. Diese lauten *Energie, Ergonomie, Fertigung, Gebrauch, Geometrie, Kinematik, Kontrolle, Kosten, Kräfte, Montage, Recycling, Sicherheit, Signale, Stoff, Termin, Transport und Wartung*. Nicht jedes dieser Kriterien ist für die Entwicklung von Prüfsystemkonzepten von nennenswerter Bedeutung, die elementaren Begriffe sind in Tabelle 2 aufgeführt und in zwei Stufen durch detaillierte *selbsterkannte* Merkmale differenziert. Das Erkennen aller relevanten Anforderungen und die daran anschließende Entwicklung von konkreten Lösungen verlangen intensive Überlegungen und zweckmäßige Ideen, die häufig spontan entstehen und nicht auf *Knopfdruck* abrufbar sind. Sehr hilfreich für diesen Prozess sind Diskussionen im Kollegenkreis, Fachbeiträge, Vorschriften, Vorbilder in der Natur, vorhandene Lösungen für vergleichbare Probleme, Experimente und besonders die eigenen Erfahrungen. In der Praxis hat sich beispielsweise die Vorgehensweise *Brainstorming* bewährt.

Jede Präzision einer Anforderung ist in der vierten Spalte der Tabelle 2 mit dem Attribut *Forderung* oder *Wunsch* versehen. Forderungen sind zwingend zu erfüllen, ansonsten ist die Gesamtlösung keinesfalls akzeptabel. Wünsche hingegen sollten nach Möglichkeit berücksichtigt werden, sofern zwischen Kosten und Nutzen ein positives Verhältnis besteht. Allgemein gilt, dass sich Anforderungen nicht gegenseitig ausschließen dürfen.

Haupt-merkmal	Vertiefung	Präzision	F/W
Energie	Energieeinleitung	Realitätsnahe Belastungen	W
Ergonomie	Mensch – Maschine – Schnittstelle	Intuitive Bedien- und Erlernbarkeit	W
		Optimaler Abstraktionsgrad	W
Gebrauch	Funktionalität	Einsatz unter extremen Umweltbedingungen	F
		Realitätsnahe Testumgebung	W
	Flexibilität	Unabhängigkeit vom Fahrzeughersteller	F
		Unabhängigkeit von der Fahrzeugklasse	F
		Unabhängigkeit von der Prüflingsausstattung	F
		Unabhängigkeit vom Entwicklungsstand	F
		Ortsveränderliche Verwendbarkeit	F
		Individuelle Gestaltung von Prüfabfolgen	W
	Effizienz	Durchführung einer zeitgerafften Erprobung	F
		Maximaler Erkenntnisgewinn	W
		Kurzfristiger Aufbau und Inbetriebnahme	W
Geometrie	Reichweite	Überbrückung der erforderlichen Distanzen	F
Kinematik	Bewegungsart	Flexible Bewegungsabläufe	F
	Geschwindigkeit und Beschleunigung	Umsetzbarkeit der vorgeschriebenen Werte	F
		Lastreserven	W
Kontrolle	Dokumentation	Kontinuierliche Messdatenaufzeichnung	W
		Lückenlose Protokollierung der Ereignisse	F
	Benachrichtigungsdienst	Unverzögliche Übermittlung von Störungen	W
Kosten	Anschaffung	Überschaubare Investitionskosten	F
	Betrieb	Minimale Betriebskosten	W

Kräfte	Gewicht	Verwendbar unter Werkstattbedingungen	F
		Genügend Traglast der Roboterarme	F
	Stabilität	Sichere Adaption der Roboterarme	F
Sicherheit	Zuverlässigkeit	Minimierung von internen Systemfehlern	W
	Zugriffsschutz	Sperren der Bedienoberfläche	W
	Arbeitssicherheit	Verhütung von Unfällen mit Robotern	F
Signale	Ein-/Ausgangssignale	Verarbeitung physikalischer Größen	F
		Verarbeitung softwarebasierter Daten	F
	Messung	Kontinuierliche Messdatenaufzeichnung	W
		Zweckmäßige Messdatenverwaltung	F
	Analyse	Manueller Zugriff auf sämtliche Signale	F
		Vergleichsmessung sämtlicher Signale	W
Wartung	Erweiterbarkeit und Modifizierbarkeit	Modulare Softwarearchitektur	F
		Umfangreiche Schnittstellen	F
		Übertragbarkeit auf anderweitige Aufgaben	W
	Verwendung von Standardkomponenten	Einsatz von Normbauteilen bei Mechanik	W
		Verwendung bewährter Software	W

Tabelle 2: Anforderungsliste

## 4.1.2 Erläuterung der Anforderungen

### 4.1.2.1 Energie

Die Realitätsnähe einer Erprobung von technischen Produkten ist in hohem Maße von der Höhe und Anzahl der Belastungen, der Kombination verschiedener Beanspruchungsarten sowie der Art und Weise von Betätigungen abhängig. Fahrzeugkomponenten sind konstruktionsseitig für eine endliche Anzahl von Betätigungen bei definierten Lastkollektiven ausgelegt. Die Einhaltung dieser Spezifikationen, also die Häufigkeit der Betätigungen und die jeweils eingeleitete Energie, ist ein entscheidendes Kriterium für die Aussagekraft der Ergebnisse. Ist die aufgebrachte Energie zum Schließen einer Tür außergewöhnlich hoch oder entstehen durch schlagartige Stöße untypische Beschleunigungen, erfolgt die Erprobung nicht realitäts- und kundennah. Analog dazu ist bei elektrischen Antrieben auf ausreichende Pausenzeiten zwischen einzelnen Ansteuerungen zu achten. Eine Überbeanspruchung verstärkt die Gefahr von Bauteilschäden, die unter realen Einsatzbedingungen nicht entstanden wären.

### 4.1.2.2 Ergonomie

Ein neu eingeführtes Prüfsystemkonzept ist im industriellen Einsatz nur effektiv, wenn es Akzeptanz durch das Bedienpersonal erlangt. Diese Akzeptanz ist abhängig von der Verständlichkeit sowie Bedien- und Erlernbarkeit. Diese Eigenschaften hängen im Wesentlichen mit der Gestaltung der Bedienoberfläche zusammen. So muss die Schnittstelle zwischen Mensch und Maschine den Kenntnisstand der Bediener berücksichtigen, im Umfeld gebräuchliche Gliederungen und Formulierungen enthalten und einen nachvollziehbaren Bezug zur umzusetzenden Aufgabe aufweisen. Das Verfahren zur Konfiguration und Abwicklung von Prüfungen soll in erster Linie durch *Parametrisieren anstelle von Programmieren* erfolgen.

Die größte Herausforderung bei der Entwicklung von Bedienoberflächen für Prüfstände ist ein optimal gewählter Abstraktionsgrad. Bei sehr oberflächlich und allgemein gehaltenen Einstellmöglichkeiten ist die Parametrierung von Prüfungen unkompliziert, die Flexibilität hingegen gering. Im Umkehrfall bieten vielfältige und tiefgehende Einstellmöglichkeiten mehr Flexibilität, die Komplexität und Gefahr von Verständnisproblemen nimmt zu. Die am besten geeignete Lösung für diese Problematik ist in enger Abstimmung mit den Versuchsmitarbeitern zu erarbeiten. Bei der erforderlichen aufgabenorientierten Schnittstelle zur Einbindung der Robotik kommt dem Abstraktionsgrad eine sehr hohe Bedeutung zu. Dieser muss gewährleisten, dass die Robotik trotz der für Versuchsingenieure ungeeigneten herstellereigenen Programmierungsumgebung einen Beitrag zur Effizienz des Prüfsystems leistet.

#### 4.1.2.3 Gebrauch

Die Durchführung eines Dauerversuchs beinhaltet den Aufbau und die Inbetriebnahme des Prüflings im Versuchsträger sowie die eigentliche Dauerbetätigung der abzusichernden Bauteile. Hinzu kommen zeitliche Intervalle für Messungen und Analysen, beispielsweise Sichtprüfungen, und Prüfungsunterbrechungen, die sich im Verlauf einer mehrwöchigen Prüfung aus einer Vielzahl von Gründen ergeben können.

Beim Gebrauch eines Prüfstands ist dessen Funktionalität von höchster Bedeutung. Die Erfahrung zeigt, dass die in den Prüfvorschriften der Hersteller geforderten extremen Umweltbedingungen, wie Temperaturen von -30 bis +80 Grad Celsius in Kombination mit einer relativen Luftfeuchte bis 95 Prozent, eine kritische Belastung für die hardwaretechnischen Komponenten der Prüfeinrichtung darstellen. Durch geeignete Maßnahmen ist sicherzustellen, dass die Robotik sowie die eingesetzte Sensorik diesen Bedingungen standhalten. Ansonsten sind lange Prüfungsunterbrechungen und steigende Kosten für zusätzliche Klimakammerlaufzeiten die Folge.

Die Gewährleistung einer realitätsnahen Testumgebung bezieht sich auf sämtliche Mechanismen und Methoden zur Betätigung eines Prüflings. Eigenschaften mechanisch beweglicher Bauteile dürfen beispielsweise nicht durch das Anbringen zusätzlicher Massen verändert werden. Die Erprobung von elektronischen Komponenten erfordert ein geeignetes Prüfumfeld, beim Fensterheber besteht dies neben dem eigentlichen Fensterhebersystem aus einem Steuergerät, originalen Kabelbäumen und einer Restbussimulation. Zur realitätsnahen Erprobung zählt gleichermaßen die kombinierte Prüfung im Gesamtsystem, wenn die jeweiligen Einzelteile in einer Beziehung zueinander stehen und Wechselwirkungen hervorrufen. Bei einer Fahrzeugtür ist es nicht zielführend, Türrohbau, Scharniere, Schloss, Fensterheber und Dichtungen als Einzelteile getrennt real zu erproben. Hier bietet die Simulation bereits zuverlässige und belastbare Ergebnisse. Treten jedoch am Türrohbau beim Zuschlag Risse an Schweißverbindungen auf, ist die Untersuchung möglicher Einflüsse auf benachbarte Bauteile und der Auswirkungen auf das Gesamtsystem unbedingt notwendig.

Neben der Funktionalität ist zur entwicklungsbegleitenden Absicherung ein besonders hohes Maß an Flexibilität gefordert. Das Prüfsystemkonzept muss unabhängig vom Fahrzeughersteller, von der Fahrzeugklasse und von der Prüflingsausstattung einsetzbar sein. Die Ausstattung eines Prüflings variiert in den Erprobungsphasen zwischen einer Rohkarosserie für die ersten Freigabeversuche und einem Gesamtfahrzeug zur serienbegleitenden Erprobung. Die mechanischen Bestandteile des Prüfsystems müssen daher so beschaffen sein, dass sämtliche Varianten mit möglichst geringem Zusatzaufwand adaptierbar sind. Dies gilt analog für die elektronische Schnittstelle zum Bordnetz. Der Verbindungsaufbau zur Fahrzeugelektronik eines jeden Herstellers ist mit überschaubarem Aufwand zu gewährleisten. Dabei ist zu be-

rücksichtigen, dass die Vollständigkeit der elektronischen Ausstattung je nach Entwicklungsstand stark variieren kann. Fehlen in frühen Entwicklungsphasen Kabelbäume und Steuergeräte, muss die Ansteuerung elektrischer Antriebe durch eine direkte Bestromung erfolgen.

Zur optimalen Nutzung von Werkstatt-, Klimakammer- und Roboterkapazitäten ist eine ortsflexible Verwendbarkeit des Prüfsystemkonzepts gefordert. Ortswechsel müssen mit möglichst geringem Aufwand durchführbar sein, daher sind grundsätzlich Schnellspannvorrichtungen und Steckverbindungen einzuplanen.

Bei der Ausarbeitung eines Prüfplaneditors zur Erstellung und Parametrierung von Prüfabläufen ist die Umsetzbarkeit der herstellerspezifischen Prüfvorschriften zu beachten, die allerdings der Geheimhaltung unterliegen und nur dem direkten Versuchsumfeld zugänglich sind. Je nach Hintergrund und Verlauf einer Prüfung muss es möglich sein, Prüfabläufe kurzfristig anzupassen.

Ein wesentlicher Erfolgsfaktor für das Prüfsystem ist die Effizienz bei der Ausführung von zeitgerafften Lebensdauertests. Neben offensichtlichen technischen Defekten besteht während der Erprobungsdauer die Gefahr von Prüfungsunterbrechungen, deren Ursachen auf den ersten Blick nicht ersichtlich sind. Gründe dafür sind sporadische Fehler am Prüfling oder Fehlfunktionen in der Prüfsystemsoftware. Sporadische Fehler am Prüfling treten erfahrungsgemäß selten auf, können aber zu einem ungünstigen Zeitpunkt, beispielsweise am Wochenende, lange Unterbrechungen verschulden. Bei unkritischen Fehlern, wie dem kurzzeitigen Ausfall eines Fensterhebers, darf die Prüfung erst bei wiederholtem Auftreten unterbrochen werden. Bei kritischen Störungen, wie Problemen beim Entriegeln einer Tür, muss der Prüfzyklus sofort stoppen. Die Anforderungen nach Realitätsnähe und Zeitraffung müssen in einem ausgewogenen Verhältnis stehen. Zur Sicherstellung der Realitätsnähe und zur Vermeidung von zeitraffungsbedingten Funktionsfehlern bedarf es einer Überwachung durch das Prüfsystem. Zusätzliche Informationen im Zusammenhang mit zeitgerafften Prüfungen sind in den Quellen [Mai-07] und [Ung-09] zu finden.

Bedingt durch die in Kapitel 2.2 beschriebene allgemeine Verkürzung der Entwicklungszeiten für Fahrzeugprojekte und die damit eng verbundene Einsparung von realen Versuchen, ist es von erheblicher Bedeutung, aus den verbleibenden Erprobungsschleifen ein Maximum an Erkenntnissen gewinnen zu können.

Die Minimierung der Aufbauzeit erfordert Hardwarekomponenten, die eine komfortable und flexible Adaptierung des Prüfstands am Prüfling ermöglichen. Erfahrungsgemäß benötigt diese überwiegend mechanische Tätigkeit viel Zeit und besitzt daher enormes Verbesserungspotential. Zur Inbetriebnahme der elektrischen Antriebe unter Einbeziehung der gesamten Fahrzeugelektronik sind Softwaretools nötig, die den Vorgang unterstützen und geeignete Analysefunktionen zur Lokalisierung von Fehlern einschließen. Der Aufwand für Aufbau und Inbetriebnahme muss in einem ausgewogenen Verhältnis zum Umfang der Prüfung und zu den dafür kalkulierten Kosten stehen. Der Aufbau einer freigaberelevanten Gesamtabstimmung darf maximal zwei Wochen in Anspruch nehmen. Sogenannte *abgespeckte* Dauerläufe an Komplettfahrzeugen mit vergleichsweise geringem Prüfumfang sollten innerhalb von zwei Arbeitstagen anlaufen.

#### 4.1.2.4 Geometrie und Kinematik

Die Ausführung von mechanischen Betätigungen wird im Zusammenhang mit dem entwickelten Prüfsystemkonzept durch Roboter realisiert. Für die Reichweite zur Betätigung der

gewöhnlich an Fahrzeugen verbauten beweglichen Karosserieelemente, beispielsweise Front- und Heckklappen oder Anschlag- oder Flügeltüren, sind erfahrungsgemäß mindestens 1.500 Millimeter gefordert. In diesem Kontext sei erwähnt, dass neben der Reichweite die optimale Positionierung des Roboterarms gegenüber dem zu bewegenden Objekt zu beachten ist. Weiterhin ist es unverzichtbar, dass die Aktoren angemessene und realitätsnahe Betätigungsformen durch Linear- und Kreisbewegungen nachbilden können. Knickarmroboter sind durch ihre freie Beweglichkeit im dreidimensionalen Raum in der Lage, diese Bewegungsmuster nachzustellen.

Hinsichtlich der Geschwindigkeiten und Beschleunigungen müssen die in den herstellerspezifischen Prüfvorschriften definierten Lastfälle uneingeschränkt realisierbar sein. Zum Erreichen einer Türschließgeschwindigkeit von 2,5 Metern pro Sekunde auf Schlosshöhe bedarf es bei einem Radius von 0,8 Metern mindesten einer Winkelgeschwindigkeit von 180 Grad pro Sekunde der Achse 1, aus welcher der Roboter diese Drehung hauptsächlich generiert. Zeitge- raffte Erprobungen erfordern grundsätzlich Kompromisse, Beispiele sind ein Verzicht auf nichtschädigende Belastungen oder eine Belastungserhöhung mit gleichzeitiger Verringerung der Testzyklen. Das Prüfsystem muss so konzipiert sein, dass der Zeitaufwand für die komplette Versuchsdurchführung im Bedarfsfall minimal ausfällt. Genügend Lastreserven bei der Auswahl der Roboterarme schaffen dazu eine Voraussetzung.

#### **4.1.2.5 Kontrolle**

Eine permanente Kontrolle und exakte Reproduzierbarkeit eines Prüfablaufs ist nur auf Basis von Messdaten gegeben. Je umfangreicher und detaillierter diese vorliegen, desto genauer sind Fehlfunktionen und Abhängigkeiten von bestimmten Faktoren analysierbar. Sporadisch ausgeführte Messungen haben bei der Dauererprobung keine repräsentative Aussagekraft. Während des automatisierten Prüfablaufs ist eine kontinuierliche Aufzeichnung der Betätigungsenergie sowie der daraus resultierenden Vorgänge anzustreben. So wird ermöglicht, plötzlich eintretende Bauteilveränderungen einem konkreten Zyklus zuzuordnen und Grenzwerte für eine Unterbrechung des Prüfablaufs einzustellen.

Die Ergebnisse einer Erprobung werden üblicherweise in Form eines detaillierten Versuchsberichts zusammengefasst. Neben einer Fotodokumentation und repräsentativen Messkurven beinhaltet dieser auch eine chronologische Abfolge aller Ereignisse während der Prüfung. Dazu gehören Bauteildefekte, Fehlfunktionen, Auffälligkeiten, Prüfungsunterbrechungen und die jeweils aktuellen Randbedingungen. Das Prüfsystem muss daher zur Nachvollziehbarkeit der Erprobung und Optimierung der Prüfeinrichtung permanent Informationen über aktuelle Zyklenstände, sämtliche Varianten von Fehlfunktionen am Prüfling, Grenzwertüberschreitungen von Sensordaten und prüfsysteminterne Störungen dokumentieren.

Im Zusammenhang mit dieser Dokumentationsfunktion ist ein Dienst wichtig, welcher zur Sicherstellung der zeitge- rafften Erprobung bei Stillständen unverzüglich Störungsmeldungen an die verantwortlichen Versuchsingenieure übermittelt.

#### **4.1.2.6 Kosten**

Für das in dieser Arbeit entwickelte Prüfsystemkonzept wurde von Beginn an ein industrieller Einsatz angestrebt. Die Kosten sind daher ein zentraler Faktor, den es bei der Umsetzung stets zu berücksichtigen gilt. Unter realistischen Annahmen muss eine Amortisation des Investitionsvolumens in einem Zeitraum von fünf Jahren erfolgen. Weiterhin sind durch geeignete

Maßnahmen die Betriebskosten der Anlage so gering wie möglich zu halten. Bei störungsbedingten oder planmäßigen Prüfungsunterbrechungen sollten große Energieverbraucher automatisch abgeschaltet werden. Der Verbrauch von kostenintensiven und verlustreichen Energieformen ist möglichst zu vermeiden. Druckluft kostet etwa den zehnfachen Preis von elektrischer Energie [Schö-11].

#### **4.1.2.7 Kräfte**

Wie bereits in Abschnitt 4.1.2.3 beschrieben, sind die Hardwarekomponenten der Prüfeinrichtung so auszuwählen und zu konstruieren, dass eine ortsflexible Verwendung ohne erheblichen Zusatzaufwand möglich ist. Schwergewichtige Bestandteile des Prüfsystems müssen daher auf Rollen oder mittels eines Hubwagens in den Versuchswerkstätten transportabel sein.

Die Roboterarme benötigen eine ausreichende Traglast, um mit den am Flansch montierten Roboterwerkzeugen Bewegungen von Karosserieteilen auszuführen. Konkrete Grenzwerte sind aufgrund der Abhängigkeit von der Armstellung und der Beschleunigung nur auf den Einzelfall und die konkret vorherrschenden Achsmomente zu beziehen. Nach Erfahrungswerten ist eine Traglast von mindestens 12 Kilogramm notwendig, um bewegliche Karosserieelemente entsprechend der Prüfvorschriften zu betätigen. Steifigkeitsmessungen mit einer Belastung von maximal 500 Newton erfordern eine Traglast von mindestens 30 Kilogramm. Eine sichere Adaption der Roboterarme am Prüfling ist eine Grundvoraussetzung.

#### **4.1.2.8 Sicherheit**

Als wesentliche Voraussetzung für entwicklungsbegleitende Versuche zur Ermittlung der Langzeitbetriebsfestigkeit gilt deren Durchführung im Zeitraffer. Nur dadurch ist es möglich, noch während des Entwicklungszeitraums Aussagen über die Haltbarkeit zu treffen und nötige Optimierungen einzuleiten. Die Zuverlässigkeit eines Prüfsystems ist daher von elementarer Bedeutung. Unterbrechungen im Prüfablauf, die ursächlich in internen Systemfehlern begründet sind, gilt es zu minimieren. Ein hierfür notwendiger Ansatz ist eine modular aufgebaute Softwarestruktur, die eine isolierte Validierung der Teilkomponenten zulässt. Zudem erhöht sich damit die Aussicht, auftretende Fehler zeitnah zu lokalisieren. Softwaretechnische Mängel, sowohl im Programmcode als auch in der Entwicklungsumgebung, lassen sich nie gänzlich ausschließen. Die Wahl einer Entwicklungsumgebung mit der Gewährleistung für eine optimale Unterstützung und Wartung ist von großer Bedeutung.

Für die Bedienung von Prüfsystemen sind Basiswissen und qualifizierte Erfahrungen der Bediener als Voraussetzung anzusehen. Änderungen an den Einstellungen durch unberechtigte Personen können zu Schäden am Prüfling führen. Der Schutz gegen unbefugten Zugriff ist daher ein relevanter Bestandteil der Bedienoberfläche. Die Arbeitssicherheit zur Verhütung von Unfällen hat beim Einsatz von Industrierobotern oberste Priorität, da aufgrund der sich bewegenden Massen bei hohen Geschwindigkeiten eine große Gefahr für Menschen im unmittelbaren Umfeld ausgeht.

#### **4.1.2.9 Signale**

Die Hauptaufgabe eines Automatisierungssystems umfasst das Einlesen, Verarbeiten und Ausgeben von Signalen. Für das Prüfsystem ist gefordert, über analoge und digitale Kanäle physikalische Größen zu erfassen. Dazu gehören Spannung, Strom, Kraft, Weg und Ge-



schwindigkeit. Des Weiteren ist es notwendig, softwarebasierte Informationen, wie Parameter aus einem Steuergerät, zu berücksichtigen. Als elementare Forderung gilt daher, Schnittstellen für physikalische und softwaretechnische Ein- und Ausgangssignale vorzuhalten.

Das Bestreben nach einer kontinuierlichen Messdatenerfassung wurde bereits in Abschnitt 4.1.2.5 erläutert. Abgesehen von eindeutigen Mängeln, wie Brüchen an Bauteilen oder Fehlfunktionen in der Elektronik, kann eine objektive Analyse und Bewertung der Qualität eines Prüflings ebenfalls nur mit Hilfe von Messdaten erfolgen. Ein Beispiel ist die Aufzeichnung des Strom- und Spannungsverlaufs bei der Fensterheberbetätigung. Als Resultat können Daten, beispielsweise der Antriebsstatus des Motors, die Fensterheberposition oder die Fehler-speichereinträge, vom Steuergerät gelesen werden. Eine fortlaufende Messdatenerfassung führt allerdings zu einer beträchtlichen Datenmenge. Für deren Verwaltung müssen Werkzeuge und Methoden zur Verfügung stehen, welche eine Analyse in praktikabler Weise ermöglichen.

Parallel zur automatisierten Messung muss eine Möglichkeit bestehen, problembezogene Daten zur Analyse von aktuell vorliegenden Störungen manuell aufzuzeichnen. Dies betrifft hauptsächlich die elektrische Inbetriebnahme von Prüfaufbauten, da hierbei gelegentlich Funktionsstörungen auftreten, die erst durch zeitaufwendige Nachforschungen behebbar sind. Ein geeignetes Tool, welches eine Vergleichsmessung sämtlicher Signale im Prüfsystem ermöglicht, ist hier zielführend. Entscheidend ist dabei der manuelle Zugriff auf sämtliche extern aufgenommenen Messwerte und prüfsysteminternen Größen.

#### **4.1.2.10 Wartung**

Für einen erfolgreichen Einsatz des Prüfsystems als Standardprüfstand bedarf es einer ständigen Aktualisierung durch technologisch angepasste Prüfmodule. Für Erweiterungen steht in der Praxis nur begrenzt Zeit zur Verfügung. Daher ist eine modulare Softwarearchitektur, durch die neue Prüfmodule zum Großteil aus vorgefertigten und bewährten Bausteinen zusammengesetzt werden können, unentbehrlich. Ausschließlich für prüfungsspezifische Besonderheiten sind individuelle Lösungen zulässig.

In der Zukunft benötigte oder verfügbare Hard- und Softwarekomponenten sind zum jetzigen Zeitpunkt nicht absehbar. Das Prüfsystem muss daher vielfältige Schnittstellen aufweisen, die eine Adaption weiterer Prüfmittel sicherstellen.

Das Prüfsystem soll nicht nur als Einzellösung für eine sehr begrenzte Anzahl von Versuchen einsetzbar sein. Vielmehr existiert die Bestrebung, die Architektur des Konzepts als Basis für eine Vielzahl von Prüfungen im Bereich der Fahrzeugabsicherung und darüber hinaus zu etablieren. Dies bezieht sich auch auf Teillösungen, beispielsweise die aufgabenorientierte Schnittstelle zur Robotik oder die Methoden zur Verwaltung der Messdaten.

Die Hardware zur Ausführung von mechanischen Betätigungen ist der anfälligste Bestandteil des Prüfsystems. Aufgrund der Forderung nach einer zeitgerafften Erprobung ist der Prüfling in vergleichsweise kurzer Zeit extremen Umweltbedingungen ausgesetzt. Diese Belastungen stellen eine große Herausforderung für die Dauerlaufeinrichtung dar, sind doch für einen wirtschaftlichen Erfolg eine Vielzahl von Dauerläufen zu absolvieren. Erfahrungen zeigen, dass regelmäßige Wartungsarbeiten sowie gelegentliche Reparaturen unvermeidbar sind und einkalkuliert werden müssen. Um die Ersatzteilkosten und Prüfungsunterbrechungen durch Instandsetzungen so gering wie möglich zu halten, sind nach Möglichkeit Standardkomponenten einzusetzen. Diese sind in der Regel ausreichend erprobt sowie kostengünstiger und zeit-

näher zu beschaffen als Sonderanfertigungen. Softwarekomponenten betrifft dies gleichermaßen, da Programme mit einer hohen Anwendungsbreite üblicherweise viele Schnittstellen aufweisen, einer umfassenden Erprobung unterliegen und herstellerseitig kompetente Unterstützung vorweisen.

### 4.1.3 Umsetzung

Im Anschluss an die Definition und Präzision der Anforderungen erfolgen in den nächsten Entwicklungsphasen der Entwurf des Prüfsystemkonzepts und die konkrete Entwicklung des Prüfstands. Hierfür ist es zweckmäßig, das Gesamtsystem in Teillösungen aufzuschlüsseln und diese zunächst isoliert bis zu einer soliden Reife zu entwickeln. Nach der Integration der Teillösungen zu einem Gesamtsystem erfolgt die Implementierung von Prüfmodulen mit anschließender Feinabstimmung und abschließender Validierung in der Praxis. Zur anschaulichen Darstellung dieses Entwicklungsprozesses hat sich im Bereich der Softwareentwicklung das V – Modell etabliert [Vmo-12].

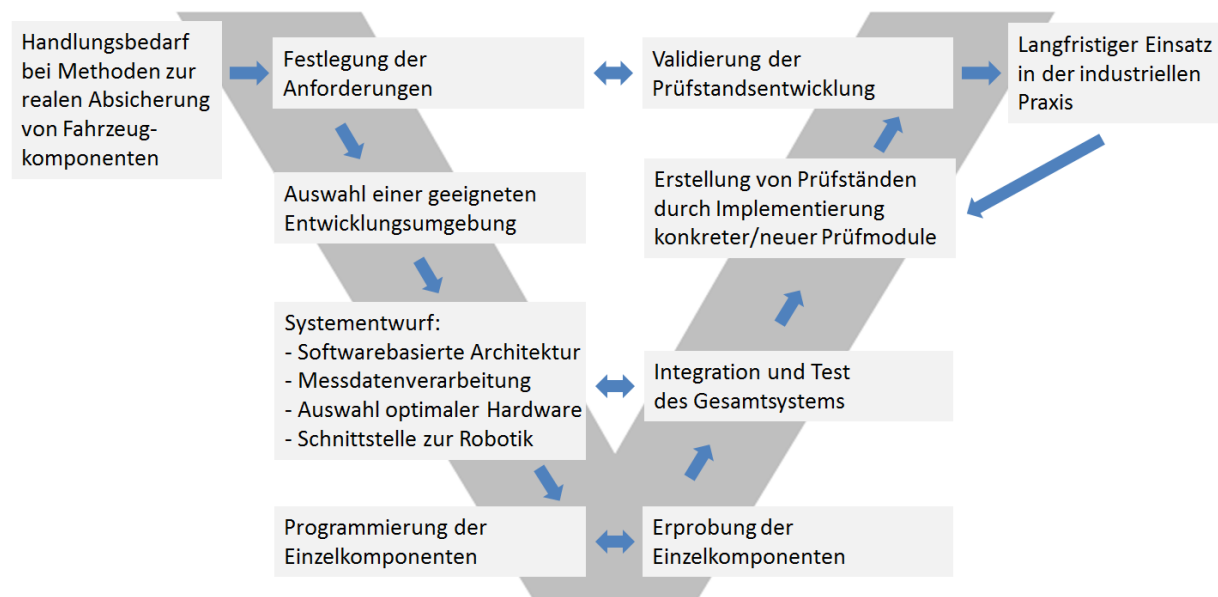


Abbildung 22: Darstellung des Entwicklungsprozesses als V – Modell

## 4.2 Verwendete Software

80 Prozent der klassischen Automatisierungssysteme arbeiteten im Jahr 2009 auf Basis einer speicherprogrammierbaren Steuerung *SPS*. Der Trend bewegt sich jedoch in Richtung des Einsatzes von Industrie – PC's. Diese sind aufgrund ihrer Arbeitsweise grundsätzlich mit herkömmlichen Büro – Computern vergleichbar, durch eine robustere Auslegung aber eher für den Einsatz im industriellen Umfeld geeignet [Her-09a]. PC – basierte Automatisierungssysteme zeichnen sich dadurch aus, dass Software allgemein leichter und schneller veränderbar ist als hardwaretechnische Systeme. Dazu kommen eine komfortable Bedienbarkeit, hohe Verfügbarkeit und geringe Kosten. Computer besitzen vielfältige Kommunikationsschnittstellen, ihre Leistungsfähigkeit steigt rasant und die Integration in die Firmeninfrastruktur ist unproblematisch. PC – Systeme sind offen für Hardwarekomponenten von Drittanbietern und dadurch flexibel verwendbar [Bur-02], [Wör-03].

Ein Nachteil von PC – basierten Systemen ist, dass diese trotz der Bemühungen vieler Firmen und Forschungseinrichtungen nicht *hart echtzeitfähig* sondern lediglich *weich echtzeitfähig* sind. Der Unterschied liegt darin, dass weiche Echtzeitsysteme alle ankommenden Eingaben *typischerweise* in der vorgegebenen Zeit abarbeiten. Harte Echtzeitsysteme, beispielsweise eine SPS, liefern Ergebnisse *immer* innerhalb einer vorgegebenen Zeit. Prinzipiell können Steuer-, Visualisierungs-, Datenverarbeitungs- und Kommunikationsaufgaben mit klassischen speicherprogrammierbaren Steuerungen oder PC – basierten Systemen ausgeführt werden. Bei dem in dieser Arbeit entwickelten Prüfsystemkonzept reicht eine weiche Echtzeitfähigkeit aus, da auf dem Prüfstandsrechner keine zeitkritischen Regelprozesse verarbeitet werden und die bei einer kontinuierlichen Messdatenerfassung vereinzelt auftretenden Ungenauigkeiten akzeptabel sind. Der Entwurf der Schnittstelle zur Robotik sieht zudem vor, sensorbasierte Regelaufgaben unabhängig vom Prüfstandsrechner auf dem hart echtzeitfähigen Robotercontroller auszuführen. Ein Schwerpunkt der softwarebasierten Systemarchitektur liegt in den Bereichen Visualisierung, Datenverarbeitung und Vernetzung. Auf Basis der beschriebenen Vorteile ist eine PC – basierte Lösung vorzuziehen [Bur-02], [Ric-04], [Tim-11].

Im Zusammenhang mit der Notwendigkeit, ein grundlegend neues Prüfsystemkonzept für Komponentenversuche zu entwickeln, ist die Frage berechtigt, ob eine in Abschnitt 3.1.1 aufgeführte am Markt verfügbare Lösung für PC – basierte Automatisierungssysteme den Anforderungen genügen kann. Beratungsgespräche mit den aufgeführten Herstellern und Testinstallationen der benannten Softwarepakete haben gezeigt, dass für die konkret formulierten Anforderungen an das Prüfsystem und die darin enthaltenen individuellen Funktionalitäten keine vorgefertigten Prüfmodule bei den Softwareherstellern zur Verfügung stehen. Für die Verwendung eines Standardprogramms wären umfangreiche Programmierarbeiten nötig, welche den Vorteil gegenüber einer kompletten Eigenentwicklung mindern. Ob die Einschränkungen, die bei einer Standardsoftware im Gegensatz zur Eigenentwicklung bestehen, gegebenenfalls zu Problemen führen, könnte erst der reale Einsatz zeigen. Nur bei einer Eigenentwicklung ist diese Gefahr komplett auszuschließen, da keine softwareseitigen Limitierungen bestehen.

Auf der Grundlage dieser Erkenntnisse ist der folgende Kompromiss die bestmögliche Lösung: Die Hauptbestandteile der Prüfsystemsoftware, bestehend aus der Prüfablaufverwaltung samt Prüfmodulen, der Messdatenverwaltung, wesentlichen Teilen der Bedienoberfläche sowie der klassischen Messdatenerfassung, basieren auf einer in LabVIEW programmierten Eigenentwicklung. Dadurch ist die Forderung nach einer größtmöglichen Unabhängigkeit und Flexibilität zur Gestaltung individueller Prüfabläufe gewährleistet. Die Schnittstelle zur Fahrzeugelektronik wird durch CANoe realisiert, da dieses Softwaretool als *Quasi – Standard* für Entwicklungsarbeiten im Bereich der Automobilelektronik betrachtet werden kann und über notwendige Schnittstellen zur Verbindung mit LabVIEW verfügt. Eine Eigenentwicklung, das zeigen die Ergebnisse der Diplomarbeit [Vog-07a], wäre in Anbetracht der herstellerspezifischen Eigenheiten zu arbeits- und kostenintensiv.

#### 4.2.1 LabVIEW

*LabVIEW* von der Firma *National Instruments* ist eine grafische Programmierumgebung, bei der die Programme nicht textbasiert, sondern durch Blockschaltbilder und Signalflussdiagramme erzeugt werden. Der Einsatz ist besonders für messtechnische Labore bestimmt, daher liegt ein Schwerpunkt in der Bedienung und Visualisierung von Automatisierungssystemen [Ler-07], [Geo-09]. Zusätzlich verfügt *LabVIEW* über umfangreiche und komfortable Mathematikfunktionen, Dateifunktionen sowie Treiber für alle gängigen Schnittstellen der Automatisierungstechnik. Im Rahmen der Umsetzung dieses Prüfsystemkonzepts werden

Verbindungen über ActiveX, Ethernet (TCP/IP, UDP und OPC) sowie RS232 hergestellt. Die Firma National Instruments bietet ein umfangreiches Sortiment messtechnischer Hardware an, deren softwaretechnische Einbindung problemlos funktioniert. Spezialkomponenten von Drittanbietern können mit Hilfe von DLL's angesprochen werden.

LabVIEW eignet sich besonders für die Programmierung von Erprobungssystemen, da die Erzeugung von parallelen Abläufen, beispielsweise die Betätigung mehrerer Türen mit gleichzeitiger Messdatenerfassung, sehr unkompliziert und übersichtlich ist. Es ist ausreichend, zwei Blockschaltbilder ohne gegenseitige Abhängigkeit in einem Blockdiagramm nebeneinander zu positionieren. Bei modular aufgebauten Programmen ist es hilfreich, dass Unterprogramme gewöhnlich auch einzeln ausführbar sind. Diese Eigenschaft spart zeitintensive Fehleranalysen [Wör-03], [Müt-09].

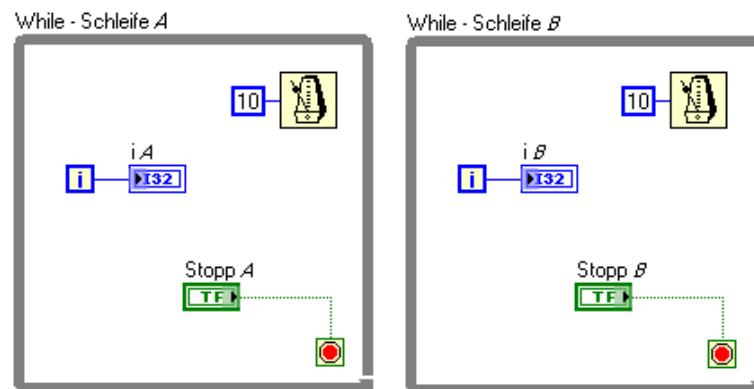


Abbildung 23: Programmcode einer parallel ablaufenden Schleifenstruktur in LabVIEW

Abbildung 23 zeigt zwei unabhängig und parallel ablaufende *While – Schleifen* mit einer Iterationsdauer von 10 Millisekunden. Die jeweils aktuellen Iterationsschritte  $i$  werden bis zur Betätigung der *Stopp – Taster* auf der Bedienoberfläche angezeigt.

Die Erstellung von Programmen erfolgt in einer Entwicklungsumgebung, an deren Ende die Erzeugung einer Programmdatei steht. Diese Programmdatei ist vom Bediener nicht mehr veränderbar und kann auf einer beliebigen Anzahl von Prüfstandsrechnern ausgeführt werden. Dafür ist lediglich die Installation der Laufzeitumgebung und Hardwaretreiber nötig. Zusätzliche Lizenzkosten fallen nicht an. LabVIEW ist weit verbreitet, die Wahrscheinlichkeit für eine zukünftige Weiterentwicklung und Unterstützung seitens des Herstellers wird dadurch erhöht. Die bewährte Software erfüllt die Anforderungen nach überschaubaren Kosten und umfangreichen Schnittstellen.

Optional gibt es mit *LabVIEW RT* eine Programmerweiterung, die unter Verwendung der dazugehörigen Hardware eine hart echtzeitfähige Umgebung bereitstellt. Damit sind auf Basis der softwarebasierten Systemarchitektur auch hochdynamische Prüfabläufe, beispielsweise die Erprobung von Verbrennungsmotoren, realisierbar. Die Entwicklungsmethodik der hart echtzeitfähigen Softwarekomponenten zeigt viele Parallelen zur Standardprogrammierung mit LabVIEW und lässt sich daher bei Bedarf mit vertretbarem Aufwand einbinden.

#### 4.2.2 CANoe

Der technische Aufbau der Fahrzeugelektronik sowie die darauf basierende informationstechnische Philosophie weisen zwischen einzelnen Automobilherstellern und deren Fahrzeugmodellen teils deutliche Unterschiede auf. Eine LabVIEW – basierte Eigenentwicklung der

Schnittstelle zwischen Prüfstandssystem und Fahrzeugelektronik wäre zu aufwendig, um den Anforderungen nach einem herstellerunabhängigen Einsatz, einer kurzfristigen Inbetriebnahme und einer zukunftsicheren Verwendung gerecht zu werden. Zudem wäre die Echtzeitfähigkeit der Restbussimulation nicht gewährleistet. Abhilfe schafft die Software *CANoe* der Firma *Vector Informatik*. Dieses Softwaretool zur Analyse, zur Simulation und zum Test von Fahrzeugbordnetzen ist weltweit verbreitet und wird heute bei jedem namhaften Automobilhersteller und Zulieferer eingesetzt. Die im Umfeld der Komponentenerprobung relevanten Bussysteme LIN, CAN, FlexRay und MOST werden allesamt unterstützt. Passende Hardware zur physikalischen Adaption ist verfügbar [Rie-07], [Vec-11b].

Die Aufgaben von CANoe im Prüfsystemkonzept sind der Datenaustausch zwischen dem Prüfstand und den Steuergeräten, die Restbussimulation bei unvollständigen Prüfaufbauten, die Aufzeichnung von Busnachrichten zu Analysezwecken sowie die Durchführung von Diagnosefunktionen auf Basis von ODX. Die zur Realisierung der Restbussimulation benötigten Daten sind äußerst wichtige Hilfsmittel bei der gesamten Elektronikentwicklung und stehen erfahrungsgemäß in optimaler Qualität beim Fahrzeughersteller zur Verfügung. Bestandteile sind eine DBC – Datei, welche der Übersetzung von Bitfolgen in leserliche Texte dient, sowie eine Programmknoten – Datei für jedes zu simulierende Steuergerät. Die Einbeziehung von Steuergeräten in den Versuchsaufbau und der Einsatz einer Restbussimulation führen, sofern reale Elektronikkomponenten nicht verfügbar sind, zu einer Steigerung der Realitätsnähe bei der Erprobung von Karosseriekomponenten.

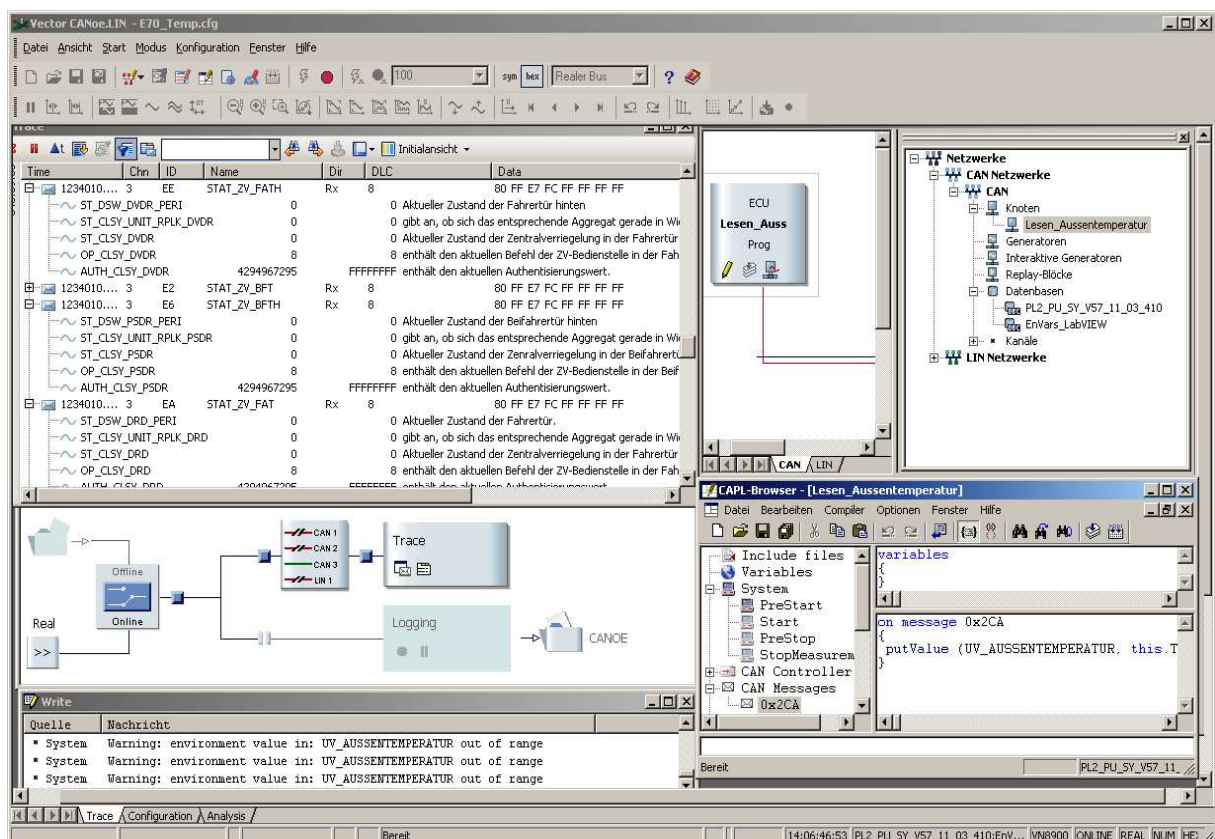


Abbildung 24: Bedienoberfläche von CANoe

CANoe bietet die Möglichkeit, den Datenfluss auf Bussystemen online in sogenannten *Trace – Fenstern* einzusehen und manuell oder in Verbindung mit dem Prüfstandssystem automatisiert aufzuzeichnen. Dadurch erweitern sich die Analysefunktionalität bei Funktionsstörungen sowie die Möglichkeit zur Qualitätsbewertung elektronischer Bauteile. Bei elektrischen Antrieben ist es seitens der Zulieferer während der Entwicklungsphase üblich, relevante Kenn-

werte kontinuierlich als Messdaten über Bussysteme auszugeben. Ebenso ist die automatisierte Ausführung von Diagnosejobs samt Speicherung der Ergebnisse durchführbar [Rie-07].

Bezogen auf die Anforderungen an das Prüfsystem stellt CANoe sehr realitätsnahe Testbedingungen für elektronische Bauteile bereit und gewährleistet eine Unabhängigkeit vom Fahrzeughersteller. Es bietet praktische Unterstützung bei der Inbetriebnahme der Fahrzeugelektronik und verspricht durch die weite Verbreitung eine hohe Zuverlässigkeit. Abbildung 24 verdeutlicht aber, dass die Anforderung an eine intuitiv bedienbare Oberfläche, die Versuchingenieuren den Umgang mit dem Prüfsystem auch bei geringer Erfahrung ermöglicht, für den durch CANoe abgedeckten Teil des Prüfsystems nicht erfüllt werden kann. Der Grund liegt in der verschachtelten Bedienoberfläche, die sich aus der umfangreichen Funktionalität von CANoe ableitet. Bedingt durch die signifikanten hersteller- und modellreihentypischen Eigenheiten der Fahrzeugelektronik, sind in der Praxis bei Aufbau und Inbetriebnahme Anpassungen mit Hilfe der C – ähnlichen Modellierungssprache CAPL notwendig. Hierfür ist die Unterstützung eines Softwareexperten mit Kenntnissen des jeweiligen Fahrzeugtyps erforderlich. Die Auswirkungen auf den Ablauf der Dauererprobung sind minimal, da Konfigurationsänderungen in CANoe nach dem Start des Dauerlaufs nicht mehr vorgesehen sind.

### 4.3 Softwarebasierte Systemarchitektur

Die Architektur eines softwarebasierten Systems beschreibt in abstrahierter Weise die darin befindlichen Komponenten sowie ihre Beziehungen zueinander. Die Qualität und Zweckmäßigkeit der Ableitung dieser Struktur aus den in Abschnitt 4.1 beschriebenen Anforderungen sind maßgeblich für die erfolgreiche Anwendung des Prüfsystems verantwortlich [Bal-00], [Sie-04], [Reu-06]. Die Art der Darstellung einer Systemarchitektur hängt von der jeweiligen Sichtweise ab.

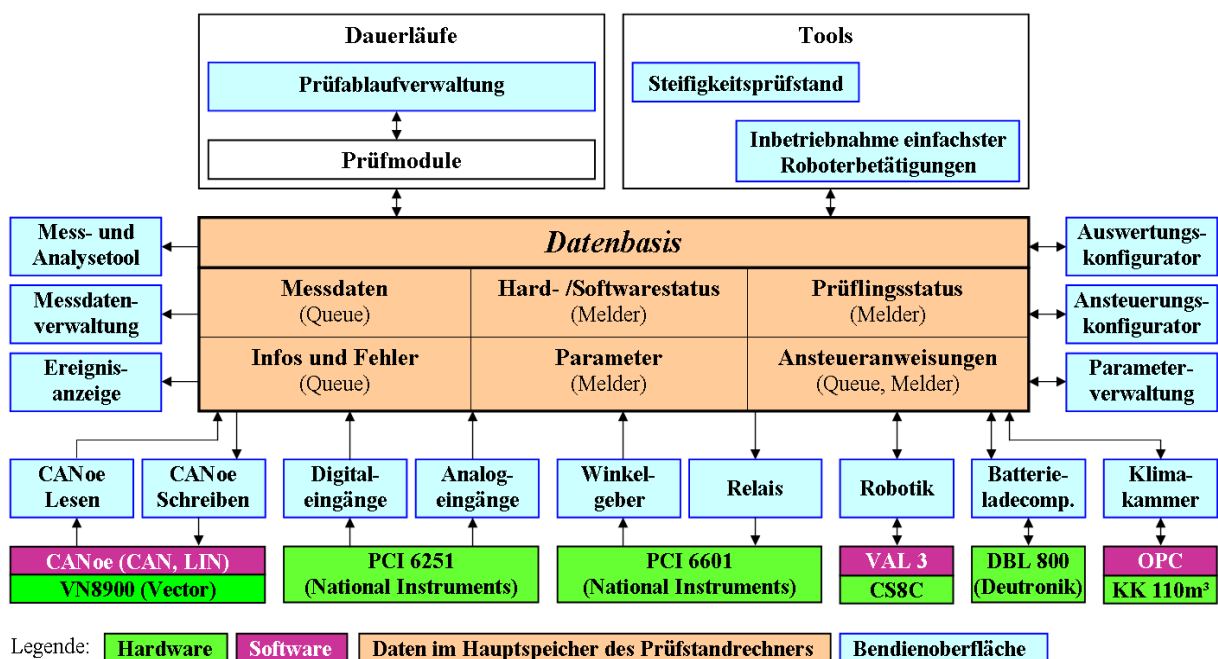


Abbildung 25: Softwarebasierte Architektur des Prüfsystemkonzepts

Die in Abbildung 25 gewählte Form zeigt primär den Datenfluss zwischen den einzelnen Komponenten sowie deren Hauptfunktion als Hard- oder Softwareschnittstelle, Datenspeicher



sowie als Bedienoberfläche zur Parametrierung und Überwachung. Diese daten- und prozess-technisch geprägte Darstellungsart soll dazu dienen, die Gliederung des Gesamtsystems in Teilkomponenten sowie die prüfsysteminternen Zusammenhänge zu verdeutlichen. Die Beschreibung der Funktionalitäten innerhalb der Komponenten erfolgt in den zugehörigen Kapiteln [Pos-04].

Tabelle 2 beinhaltet eine Reihe von *Forderungen* und *Wünschen*, die den Entwurf der Systemarchitektur maßgeblich beeinflussen. Diese sind die Flexibilität bezüglich der Prüflingsausstattung und des Fahrzeugentwicklungsstands, die Effizienz bei der Inbetriebnahme, die Zuverlässigkeit durch eine Minimierung interner Systemfehler, der Zugriff auf sämtliche internen Signale zur Analyse sowie die Modularisierung zum Zweck einer optimalen Erweiterbarkeit. Die letztgenannte Forderung spiegelt sich insbesondere in kleinen und überschaubaren Basis- und Unterprogrammen wider, die in der Entwicklungsumgebung isoliert ausführbar sind. So werden die Absicherung und Fehlersuche bei der Programmierung erleichtert.

Die Ausführung aller Komponenten innerhalb der Softwarearchitektur läuft parallel, einzig die Prüfmodule werden bei deren Aufruf sequentiell von der Prüfablaufverwaltung gestartet. Abbildung 25 beschreibt die Maximalausstattung des Prüfsystems. Im Einzelfall lassen sich nicht benötigte Hard- oder Softwareschnittstellen beim Programmstart deaktivieren.

Die Erfahrung hat gezeigt, dass sich die Ansprüche an Prüfabläufe von Komponentenversuchen stetig verändern und erweitern. Es ist mit vertretbarem Aufwand nicht möglich, eine übersichtliche und intuitiv bedienbare Oberfläche zu entwerfen, die alle Eventualitäten der zukünftigen Erprobung berücksichtigt. Das Konzept sieht an dieser Stelle eine strenge Trennung zwischen *Softwareexperten* und *Bedienern* vor. Sollten im Rahmen eines Erprobungsauftrags Erweiterungen am Prüfsystem notwendig werden, wie besondere Messungen, alternative mechanische Betätigungen oder zusätzliche Sensoren, sind diese als zusätzliche Option ausschließlich durch Softwareexperten zu implementieren. Durch jede dieser Ergänzungen erhöht sich die Funktionalität und Flexibilität des Gesamtsystems auf Dauer hin zu einer Prüfsystemsoftware mit umfangreicher und praxisbezogener Ausstattung. Der Begriff *Softwareexperte* steht im weiteren Verlauf dieser Arbeit als Synonym für die Personen, die mit der Wartung und Weiterentwicklung des Prüfsystems betraut sind.

#### 4.3.1 Datenbasis

Die Datenbasis ist aus datenflusstechnischer Sicht die zentrale Komponente im Prüfsystemkonzept. Zur Laufzeit sind hier alle systeminternen Zustände hinterlegt oder Daten werden über die Datenbasis zwischen Sender und Empfänger ausgetauscht. Ein Inhalt ist der aktuelle Prüflingsstatus, welcher vom Auswertungskonfigurator auf Basis des Hard- und Softwarestatus generiert wird. Hinzu kommen Ansteueranweisungen, die vom Ansteuerungskonfigurator nach einer Veranlassung durch die Prüfmodule in Richtung der Aktorik fließen. Sämtliche Parameter zum Prüfablauf sind ebenfalls in der Datenbasis gespeichert, diese lassen sich von jeder Komponente im Prüfsystem auslesen oder überschreiben. Die im Verlauf von Erprobungen anfallenden Messdaten, Informationen und Fehlermeldungen gelangen über die Datenbasis zur weiterverarbeitenden Komponente.

Die Speicherung dieser Daten zur Laufzeit erfolgt direkt im Hauptspeicher des Rechners, dadurch sind minimale Zugriffszeiten garantiert. Die Umsetzung in LabVIEW geschieht durch *Melder-* und *Queuefunktionen*. Ist nur der aktuelle Status eines Signals bedeutsam, kommen Melder zum Einsatz. Dies kann der Istwert an einem analogen Eingang sein. Eine Queue, auch Warteschlange genannt, ist notwendig, sobald die Nichteinhaltung des Echtzeitablaufs

zu kritischen Zuständen führen kann, weil beispielsweise ein Relais eingeschaltet wurde, aber nicht mehr zurückgesetzt wird. Queues stellen durch eine sequentielle Abarbeitung die korrekte Ansteuerungsreihenfolge ohne die Gefahr von Informationsverlusten sicher. Einzelne Queues besitzen parallel einen Melder, welcher den Istzustand einer Ansteuerung auch über die Verarbeitungsdauer hinaus in der Datenbasis bereitstellt.

Aus einer Abwägung zwischen Messgenauigkeit und vertretbarem Datenvolumen ergibt sich für Prüfsysteme in der Karosserieerprobung eine interne Aktualisierungsrate von 1.000 Hertz. Damit können zum Beispiel auch Reaktionszeiten von Karosseriesteuergeräten auf Änderungen an den Eingängen ausreichend genau gemessen und mit den Anforderungen der Lastenhefte verglichen werden. Vergleichbare Prüfstände in der Motorenentwicklung arbeiten überwiegend mit einer Abtastrate von 10 Kilohertz [Bon-10]. Kostengünstige externe Sensoren, die für den Einsatz unter extremen Klimabedingungen ausgelegt sind, weisen oftmals erhöhte Streuungsmaße auf. Die Sollvorgabe für Messungen an Karosseriekomponenten beträgt typischerweise 100 Hertz. Durch die Einstellung einer zur Sollvorgabe erhöhten Abtastrate eröffnet sich die Möglichkeit des Einsatzes von Filter- und Statistikfunktionen zur Optimierung der Kurven, ohne eine kostenintensivere Sensorik zu benötigen. Einkanalige Strommessungen von zehn Sekunden erzeugen bei einer optimalen Speichermethode ein Datenvolumen von 80 Kilobyte. Diese Datenmenge ist in Anbetracht der heute zur Verfügung stehenden Festplattenkapazitäten akzeptabel.

Maßgeblich verantwortlich für die interne Aktualisierungsrate ist nicht die Datenbasis, sondern sind die Komponenten der Hard- und Softwareschnittstellen. Diese müssen aktualisierte Daten mit entsprechend kurzen Latenzzeiten in die Datenbasis übertragen. Aus technischen Gründen ist das nicht bei allen Komponenten möglich oder sinnvoll. Weitere diesbezügliche Hinweise folgen im Verlauf der Arbeit in den entsprechenden Kapiteln. Der interne Lese- und Schreibzugriff auf die Datenbasis wurde an einem Entwicklungsrechner *IBM T500, 2x2,8 Gigahertz Intel Centrino 2* unabhängig vom Datentyp im Durchschnitt mit 2,3 Mikrosekunden berechnet und ist damit geschwindigkeitstechnisch ausreichend.

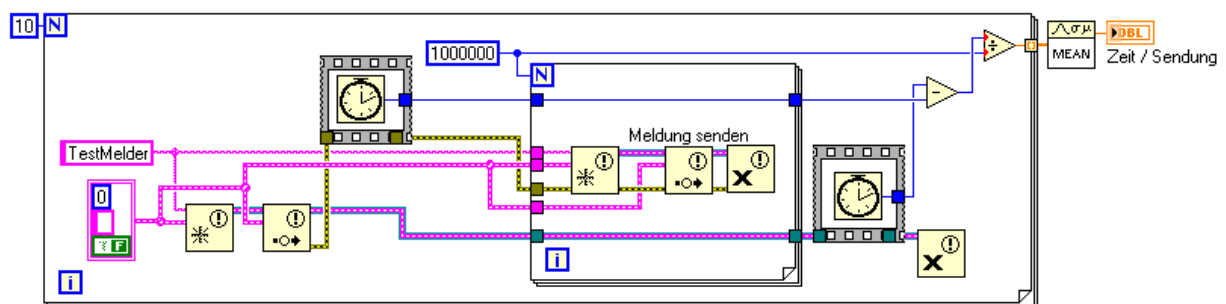


Abbildung 26: Programmcode zur Zeitmessung des Zugriffs auf die Datenbasis

Abbildung 26 beschreibt ein LabVIEW – typisches Testprogramm zur Messung der Programmverarbeitungszeit. Im konkreten Fall wird die mittlere Zugriffszeit auf die *Datenbasis* über den Schreibvorgang *Meldung senden* berechnet. Nach einer Million Aufrufen der dazu benötigten Funktionsabfolge in der inneren *For – Schleife* erfolgt die Subtraktion der Systemzeiten vor und nach den Schreibvorgängen mit anschließender Division durch die Iterationen. Die äußere *For – Schleife* sorgt für eine zehnmahlige Versuchsdurchführung. Am Ende wird der statistische Mittelwert aller Versuchsdurchläufe ausgegeben. Das Prinzip dieses Aufbaus dient in vielen Fällen als Entscheidungsgrundlage, um zwischen verschiedenen Varianten den Programmcode mit der besten Performance zu ermitteln.

Über das Mess- und Analysetool, beschrieben in Abschnitt 4.3.7, besteht ein direkter Zugriff auf sämtliche Melder und somit auf die aktuellen internen Zustände im Prüfsystem. Der



Wunsch nach Vergleichsmessungen sämtlicher Signale zu Analysezwecken ist damit erfüllt. Ein Stromausfall oder Systemabsturz bedeutet keine Gefahr für die Inhalte der Datenbasis. Die Parameter sind über die Parameterverwaltung auf der Festplatte gespeichert, beim Neustart des Programms erfolgt deren direkte Übertragung in die Datenbasis. Die Inhalte der übrigen Melder werden im Millisekundentakt neu berechnet und aktualisiert. Lediglich zum Ausfallzeitpunkt nicht verarbeitete Messdaten, Dauerlauf – Informationen und Fehlermeldungen gehen verloren.

### 4.3.2 Hard- und Softwareschnittstellen

Die primäre Funktion der Hard- und Softwareschnittstellen besteht in der Aufgabe, Daten von Hardwarekanälen oder externen Softwareprogrammen einzulesen und ohne Zeitversatz in die Datenbasis zu überführen. Ferner sind Ansteueranweisungen oder Zustandsänderungen direkt aus der Datenbasis an die Empfänger zu übertragen.

Zur Erhöhung der Flexibilität im praktischen Einsatz ist die Prüfsystemsoftware grundsätzlich auf jedem LabVIEW – fähigen Rechner, vom Industrie – PC bis zum Standardnotebook, lauffähig. Die Hardwareschnittstellen Ethernet oder RS – 232 sind gewöhnlich an beiden Rechnersystemen vorhanden, die Anschlussmöglichkeiten für messtechnische Hardware variieren in Abhängigkeit von der Rechnerarchitektur. Typischerweise sind heute PCI, PCIe, PXI oder USB im Einsatz. Für jeden einzelnen Analog- oder Digitalkanal ist die Zuordnung einer Hardwareressource in der Prüfsystemsoftware einstellbar und im laufenden Betrieb änderbar.

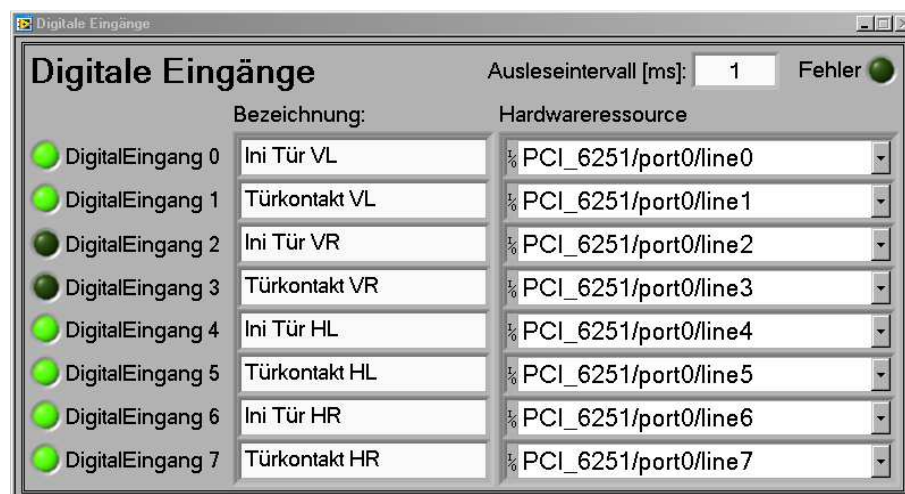


Abbildung 27: Auswahl verfügbarer Hardwareressourcen in der Komponente *Digitale Eingänge*

Die Funktionalität einer Softwareschnittstelle hängt in erster Linie von der Kompatibilität der beteiligten Programme zueinander ab. Sind dafür zusätzliche Softwarekomponenten, wie zum Beispiel *ActiveX* oder *.NET*, nötig, die von vielen Programmiersprachen eingebunden werden können, erhöht sich die Gefahr von Kompatibilitätsproblemen. Ein Grund liegt darin, dass die Softwarehersteller durch diverse Kombinationsmöglichkeiten und stetig neue Programmversionen nicht alle theoretisch möglichen Zusammensetzungen ausreichend testen können. Insbesondere sporadische und nicht nachvollziehbare Betriebsstörungen wirken sich negativ auf die Laufzeit von Prüfungen aus. Softwareschnittstellen sind daher vor ihrem Einsatz in Prüfsystemen ausgiebig im praxisnahen Kontext zu erproben.

Zur Nachvollziehbarkeit des Datenflusses wird der aktuelle Zustand eines jeden Hardwarekanals und aller Softwarevariablen auf der Bedienoberfläche der jeweiligen Komponente im

Prüfsystem angezeigt. Bei Ausgängen besteht zusätzlich die Option einer manuellen Zustandsänderung.

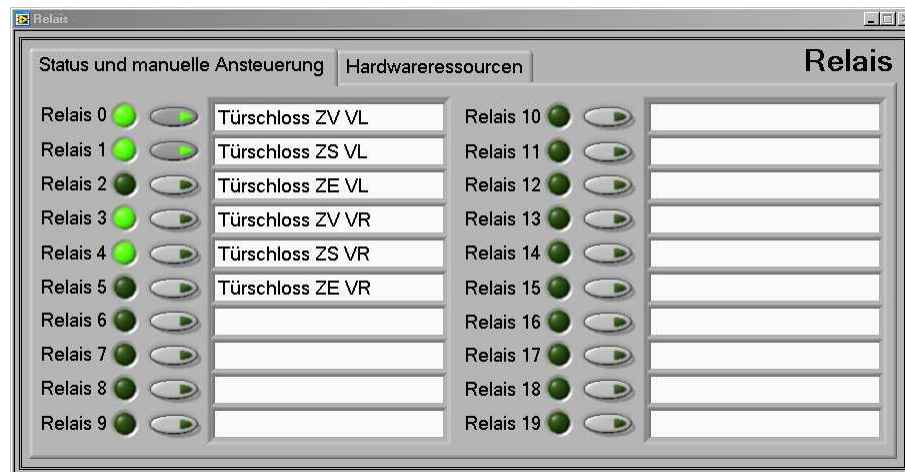


Abbildung 28: Manuelle und automatische Ansteuerung in der Komponente *Relais*

Die Forderung nach einem manuellen Zugriff auf sämtliche Signale innerhalb des Prüfsystems ist hiermit erfüllt. Daraus ergibt sich die Option, ursprünglich automatisierte Prozesse von Hand auszuführen, was sich in der Praxis als hilfreich bei der Analyse von Fehlfunktionen erwiesen hat.

#### 4.3.2.1 Analoge Eingänge

Das Einlesen der analogen Spannungseingänge wird durch die Komponente *Analoge Eingänge* realisiert. Vor einer Übertragung der Eingangsspannung in die Datenbasis erfolgt die Umrechnung in eine dem Sensor angepasste Größe und Einheit. Die Vorschrift zur Umrechnung ist in einer sogenannten Skalierung gespeichert. Skalierungen können für die gesamte LabVIEW – Umgebung einheitlich mit Hilfe des *Measurement & Automation Explorers* erstellt und geändert werden. Die Umrechnung ist durch Linearfunktionen, Polynomfunktionen oder Wertetabellen möglich. Die Aktualisierung der analogen Eingangssignale in der Datenbasis erfolgt mit 1.000 Hertz.

#### 4.3.2.2 Digitale Kanäle

Zum Anschluss digitaler Messtechnik und zur Ansteuerung elektrischer Antriebe über Relais verfügt das Prüfsystem über digitale Ein- und Ausgänge sowie Zählerkanäle. Änderungen an digitalen Eingängen werden mit 1.000 Hertz erfasst und direkt in die Datenbasis übertragen. Ein Hardwarezugriff auf die Ausgänge erfolgt unmittelbar bei einer Zustandsänderung, welche ebenso mit einer Rate von 1.000 Hertz überwacht wird.

Für die kontinuierliche Messung eines Winkelmaßes und der daraus abgeleiteten Winkelgeschwindigkeit mit Hilfe von Drehwinkelgebern stehen Zählerkanäle zur Verfügung. Bei typischen Bewegungsabläufen von Karosserieelementen, beispielsweise Türen und Klappen, ist eine maximale Winkelgeschwindigkeit von 3 Rad pro Sekunde zu erwarten. Der verwendete Typ Drehwinkelgeber gibt 3.600 Impulse pro Umdrehung aus. Daraus ergibt sich eine maximale Änderungsrate an den Zählerkanälen von 10,8 Kilohertz. Die Abtastung dieser Eingänge erfolgt durch einen internen Prozessor auf der Messkarte *NI PCI 6601* im Prüfstandsrechner mit einer festen Rate von 20 Megahertz und ist daher ausreichend [Nat-09c].

Die Abfrage des Winkelmaßes und die Übertragung in die Datenbasis durch die Softwarekomponente *Winkelgeber* erfolgt mit einer Rate von 1.000 Hertz. Auf Basis des Winkelmaßes  $\alpha$  wird durch eine numerische Differenzierung die approximierte Winkelgeschwindigkeit  $\alpha'$  errechnet. Dies geschieht mittels des *rückwärts genommenen Differenzenquotienten*, da aus dieser Variante keine zusätzliche Verzögerung bei der Übertragung in die Datenbasis resultiert.

$$\alpha' \approx \frac{\alpha(t) - \alpha(t - \Delta t)}{\Delta t}$$

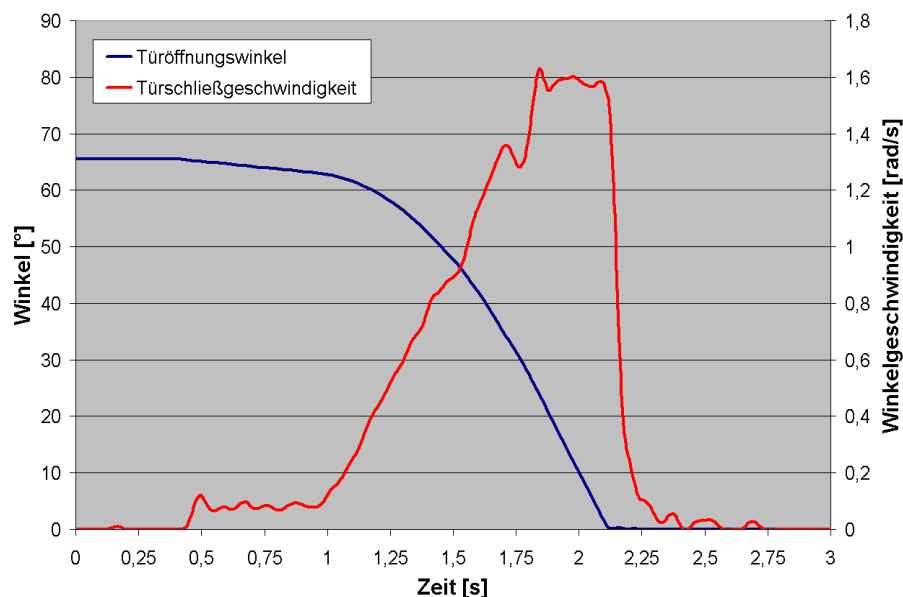
**Formel 1: Rückwärtsdifferenz zur numerischen Berechnung der Winkelgeschwindigkeit**

Ausschlaggebend für die Berechnung einer möglichst genauen Winkelgeschwindigkeit  $\alpha'$  ist eine zweckmäßige Auswahl des Zeitintervalls  $\Delta t$ . Bei numerischen Berechnungen ist stets ein geeigneter Kompromiss zwischen Abtastrate und Genauigkeit zu finden. Ist das Zeitintervall  $\Delta t$  zu kurz, kommt es zur sogenannten *Auslöschung* durch die Subtraktion von zwei gleich großen Winkelmaßen  $\alpha$ . Bei einem zu großen Wert sind Ungenauigkeiten durch eine stark ausgeprägte Mittelwertbildung die Folge. Ein Minimalwert  $\Delta t_{\text{Min}}$  für das Zeitintervall lässt sich mit der mindestens zu messenden Winkelgeschwindigkeit  $\alpha'_{\text{Min}}$  sowie der Kennwerte des eingesetzten Drehwinkelgebers berechnen.

$$\Delta t_{\text{Min}} = \frac{1}{\alpha'_{\text{Min}} \cdot \frac{\text{Impulse}}{\text{Umdrehung}}}$$

**Formel 2: Minimale Schrittweite zur numerischen Berechnung der Rückwärtsdifferenz**

Ausgehend davon, dass bei der Prüfung von Karosseriekomponenten Winkelgeschwindigkeiten ab 0,05 Rad pro Sekunde messbar sein sollen, ergibt sich bei dem oben genannten Typ Drehwinkelgeber ein minimales Zeitintervall  $\Delta t_{\text{Min}}$  von 5,56 Millisekunden für eintretende Änderungen des Winkelmaßes  $\alpha$ . Basierend auf dieser Berechnung sowie diversen Versuchen hat sich in der Praxis ein Zeitintervall  $\Delta t$  von 10 Millisekunden bewährt. Die Winkelgeschwindigkeit wird mit einer Aktualisierungsrate von 100 Hertz in die Datenbasis überführt.



**Abbildung 29: Winkelmaß und differenzierte Winkelgeschwindigkeit**

Zur Vereinfachung der Montage ist der verwendete Drehwinkelgeber Typ ein Inkremental- und kein Absolutwertgeber. Dieser Typ erfasst nur Drehwinkeländerungen, was die Implementierung einer Methode zur Referenzierung erforderlich macht. Für die Messung des Türöffnungswinkels bedeutet dies beispielsweise, dass nach dem Eintreten des Prüflingsstatus *Tür geschlossen* und einer Wartezeit von einer Sekunde zum Ausschlagen der Tür der Türöffnungswinkel auf null gesetzt wird.

#### 4.3.2.3 Robotik

Die Verbindung zur Robotik ist die umfangreichste Schnittstelle im Prüfsystem. Die Komponente *Robotik* ist abhängig vom verwendeten Robotercontrollertyp und wird beim Programmstart nach entsprechender Auswahl geladen. Für eine mechanische Betätigung mit Hilfe eines Roboters übermitteln die Prüfmodule aufgabenorientierte Befehle, zum Beispiel *Tür entriegeln*, an die Komponente Robotik. Diese verbindet sich via Ethernet und TCP/IP mit dem Robotercontroller und übermittelt die zur Ausführung der Betätigung benötigten Parameter. Ist der Bewegungsablauf beendet, wird eine Bestätigung der erfolgreichen Ausführung an die Prüfmodule zurückgegeben. Eine detaillierte Beschreibung der internen Vorgänge dieser Komponente erfolgt im Abschnitt 4.6.

Messdaten, die während der Bewegung des Roboterarms erfasst werden, gelangen über die Komponente Robotik in die Datenbasis. Diese beinhalten die aktuelle Position des Werkzeugs im Roboterkoordinatensystem sowie die Werte eines im Roboterwerkzeug integrierten Kraftsensors. Inklusiv einiger Steuerzeichen erwächst daraus eine maximale Datenmenge von 40 Byte pro Botschaft. Bei einer Abtastrate von 1.000 Hertz beläuft sich die Datenrate auf 40 Kilobyte pro Sekunde. Bei einer Übertragungsgeschwindigkeit der Netzwerkverbindung von 10 Megabit pro Sekunde, abzüglich eines *Overheads* von 36 Prozent bei einer derart geringen Nutzdatenmenge, ergibt sich bei einer Direktverbindung ohne Kollisionen eine mögliche Nutzdatenübertragungsrate von zirka 750 Kilobyte pro Sekunde. Die Netzauslastung liegt daher selbst im Extremfall einer permanenten Datenübertragung bei fünf Prozent [Sta-97].

Um eine Übertragung der Messdaten vom Roboter in die Datenbasis ohne Zeitversatz zu gewährleisten, muss der sogenannte *Nagle – Algorithmus* im Betriebssystem auf dem Robotercontroller deaktiviert sein. Dieser Algorithmus optimiert den Datendurchsatz über eine Zusammenfassung der Nutzdaten mehrerer Botschaften auf eine optimale Nutzdatenmenge von bis zu 1.500 Byte. Der Overhead verringert sich so auf zwei Prozent. Ausgehend von 40 Byte pro Botschaft würde dies eine Zusammenfassung von 37 Messdatenbotschaften zu einer Botschaft mit 1480 Nutzdatenbyte bedeuten. Die Aktualisierung der Datenbasis reduziert sich dann auf nur noch 27 Hertz [Bra-89].

#### 4.3.2.4 Batterie – Ladecomputer

Die Erfahrung hat gezeigt, dass viele Fehlfunktionen elektronischer Komponenten während der Dauererprobung mit der Energieversorgung des Fahrzeugbordnetzes zusammenhängen. Für die Laufstabilität ist es immens wichtig, den Status der Energieversorgung fortlaufend zu überwachen, sonst würde etwa bei einer Unterversorgung der Türkontakt nicht mehr korrekt ausgewertet. Der Versuch würde automatisch anhalten. Der zur Sicherstellung und Überwachung der Bordnetzspannung eingesetzte Batterie – Ladecomputer *DBL800* der Firma *Deutronic* verfügt über eine RS – 232 Schnittstelle. Über diese ist es möglich, die aktuelle Batteriespannung auszulesen.

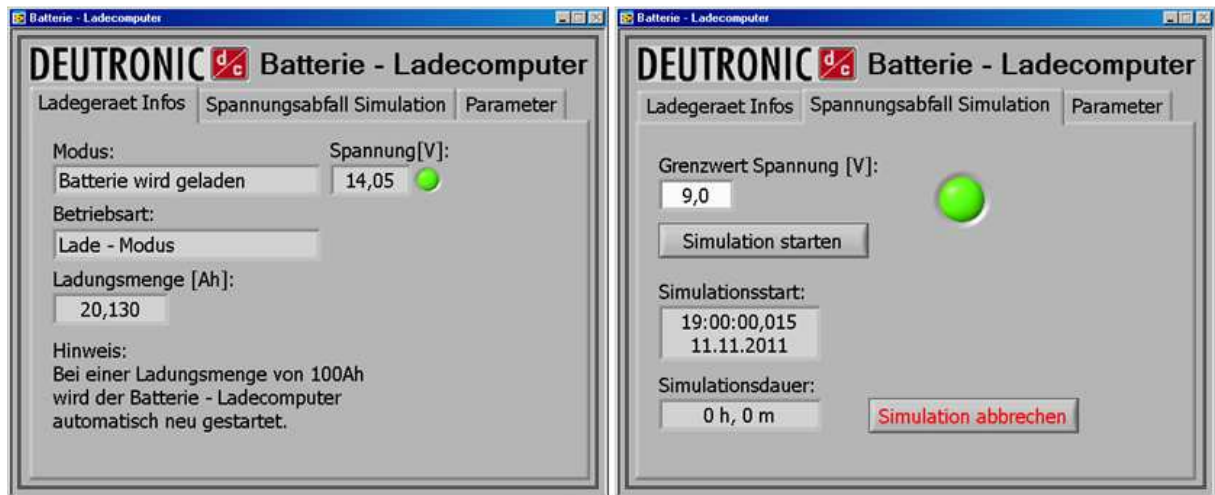


Abbildung 30: Bedienoberfläche der Komponente *Batterie - Ladecomputer*

Die Komponente *Batterie – Ladecomputer* vollzieht diesen Auslesevorgang kontinuierlich, allerdings wird bedingt durch die Antwortzeit des Ladecomputers von zirka 20 Millisekunden nur eine Aktualisierungsrate der Datenbasis von 50 Hertz erreicht. Durch eine als Puffer eingesetzte Autobatterie reicht diese Rate zu Überwachungs- und Messzwecken aus.

Für eine realitätsnahe Erprobung ist es erforderlich, gezielt Grenzsituationen, wie Über- oder Unterspannung, nachzustellen und die Reaktion des Prüflings zu dokumentieren. Dazu kann mit Hilfe der Komponente *Batterie – Ladecomputer* eine Spannungsabfallsimulation durchgeführt werden. Währenddessen pausiert der Lade- oder Ladungserhaltungsvorgang der Autobatterie, bis die Spannung auf einen definierten unteren Grenzwert abgefallen ist. Die Dauererprobung wird normal fortgesetzt. Anschließend erfolgt die Wiederaufnahme des Ladevorgangs. Fehlfunktionen bei der Ansteuerung elektronischer Komponenten, beispielsweise Reversierer am Fensterheber, werden zusammen mit dem aktuellen Spannungswert dokumentiert.

#### 4.3.2.5 Umweltsimulation

Die Hälfte einer Dauererprobung findet üblicherweise unter extremen Umweltbedingungen statt. Durch dadurch bedingte größere Ausfallrisiken verlängert sich die Verweilzeit der Prüflinge in der Klimakammer im Vergleich zu Raumtemperaturbedingungen. Anlagen zur Umweltsimulation sind aus diesem Grund mit dem Prüfsystem vernetzt, beispielsweise über Ethernet und OPC. So lassen sich äußere Bedingungen, wie Temperatur, Luftfeuchte oder Schwingungsprofil, zum Fehlerzeitpunkt dokumentieren. Die Erklär- und Reproduzierbarkeit von Fehlfunktionen steigt deutlich. Überdies werden Betriebskosten eingespart, da bei planmäßigen oder störungsbedingten Prüfungsunterbrechungen automatisch eine sofortige Abschaltung der Gerätschaften mit hohem Energieverbrauch veranlasst wird. Die Antwortzeiten der Regler sind mit zirka 50 Millisekunden vergleichsweise langsam, in Anbetracht der geringen Änderungsraten bei Temperatur oder Luftfeuchte aber ausreichend. Die Daten werden von der Komponente *Klimakammer* zwischen Regler und Datenbasis ausgetauscht.

#### 4.3.2.6 Mobiltelefon

Mit Hilfe von GSM – fähigen Mobiltelefonen ist die automatisierte Übermittlung von Statusnachrichten des Prüfsystems via SMS möglich. Seitens des Herstellers muss dafür ein Treiber zur Verfügung stehen, welcher das Mobiltelefon als Modem im Betriebssystem des Prüf-

standsrechners emuliert. Über dieses virtuelle Modem können mit Hilfe von Modembefehlen Textnachrichten verschickt werden. In der Praxis hat sich diese Methode unter Verwendung eines *Nokia 6021* bewährt. Die Hauptfunktion *+CMGS* erfordert vor dem Einsatz eine Prüfung auf einwandfreie Funktion, da diese nicht mit jedem Mobiltelefon kompatibel ist. Die Parametrierung der SMS – Funktion erfolgt über die Komponente *Ereignisanzeige*.

#### 4.3.2.7 CANoe

Zum Datenaustausch mit externen Programmen stehen in CANoe sogenannte *Umgebungsvariablen* zur Verfügung. In der Prüfsystemsoftware sind diese in einer separaten *CANdb* – Datei mit Namen, Datentyp und Standardwert gespeichert. Für den Datenabgleich zwischen den Umgebungsvariablen und der Datenbasis stehen je nach Hardware – Interface der Firma *Vector Informatik* zwei Varianten mit unterschiedlichen Latenzzeiten zur Auswahl. Die erste Variante ist die Verwendung des in CANoe integrierten *COM Servers*, über den mit Hilfe von *ActiveX* – Funktionen der externe Zugriff auf die Umgebungsvariablen möglich ist. Da der *COM* – Server nicht im hochprioren Teil der Software verankert ist, kann der Zeitversatz zwischen einer Signaländerung auf dem Fahrzeugbus und der Übertragung in die Datenbasis 50 bis 60 Millisekunden betragen [Vector Hotline]. Eine schnellere Alternative bietet das auf der Ethernetschnittstelle und dem *UDP* aufsetzende *FDX Protokoll*. Dies greift zum Datenabgleich direkt auf die Umgebungsvariablen im Echtzeit – Kernel der Hardwareschnittstelle zu und garantiert Reaktionszeiten von einer Millisekunde [Vec-10]. Die zweite Variante ist aber durch die benötigte Ethernetschnittstelle nur bei Geräten der Serie *VN8900* verfügbar.

Für den Zugriff auf die Fahrzeugbusse oder auf interne Daten der Restbussimulation müssen die Umgebungsvariablen mit Hilfe sogenannter *CAPL – Knoten* in den Simulationsaufbau integriert werden. Die Arbeitsweise dieser Netzknoten ist ereignisgesteuert: Bei einer Änderung von prüfsystemrelevanten Signalen auf den Datenbussen erfolgt eine Aktualisierung der Umgebungsvariablen. Im umgekehrten Fall werden bei Änderungen von Umgebungsvariablen durch die Prüfsystemsoftware Nachrichten auf Fahrzeugbusse geschrieben oder Busmitschriften gestartet. Für die Bedienung von CANoe und die Anpassung an neue Fahrzeugtypen sind fundierte Kenntnisse der *C* – ähnlichen Programmiersprache *CAPL* notwendig. Diese Arbeitsschritte sind den Softwareexperten vorbehalten.

#### 4.3.3 Parameterverwaltung

Die Komponente *Parameterverwaltung* dient dazu, sämtliche einstellbaren Parameter an einer zentralen Stelle im Prüfsystem zu verwalten. Beim Start des Prüfsystems werden die Parameter von der Festplatte in die Datenbasis geladen, was eine Zugriffszeit von zirka 2,3 Mikrosekunden garantiert. Über die Bedienoberfläche lassen sich die Parameter manuell ändern. Eine automatische Änderung durch die Prüfmodule ist ebenfalls möglich.

Die Speicherung der Parameter auf der Festplatte des Prüfstandsrechners erfolgt in einem speziellen betriebssystemunabhängigen Format für Konfigurationsdateien. Jeder einzelne Wert wird mit einem individuellen Schlüssel versehen. Beliebig viele Parameter lassen sich so ohne Datenverlust durch Softwareexperten zum Prüfsystem hinzufügen oder entfernen. Der interne Aufbau ist aus Gründen der Übersichtlichkeit nach Datentypen aufgeteilt. Das Hinzufügen oder Entfernen von Parametern benötigt wenige Minuten. Alle Parameter können gesammelt in einer Datei gespeichert oder aus einer Datei geladen werden. Bei wiederkehrenden Prüfungen ist der Zugriff auf altbewährte Parametersätze möglich.





Abbildung 31: Bedienoberfläche der Komponente *Parameterverwaltung*

4.3.4 Auswertungskonfigurator

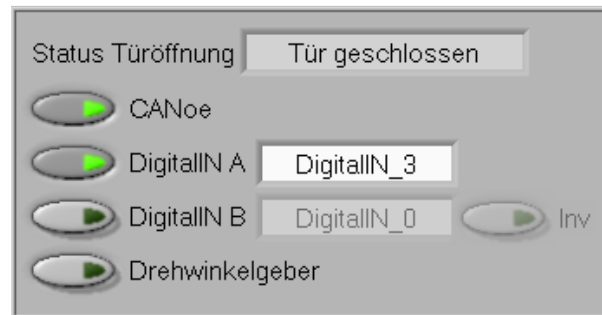
Die Komponente *Auswertungskonfigurator* hat einen essenziellen Anteil an der Flexibilität des Prüfsystems. Sie sichert die einfache Adaption von Prüflingen verschiedenster Entwicklungsstände und erleichtert die Einstellung unterschiedlicher Prüftiefen in Abhängigkeit zum Entwicklungsstand. Die Komponente abstrahiert den Status des Prüflings auf Basis der jeweils eingestellten Hardwarekanäle oder Softwarevariablen.

Zur Erklärung eignet sich das Beispiel der Ermittlung des Prüflingsstatus *Türöffnung*. Dieser kann die im Prüfsystem vordefinierten Zustände *Tür geschlossen*, *Tür offen* oder *undefiniert* annehmen. Die Ermittlung dieses Zustands basiert auf einzelnen oder miteinander kombinierten Eingangsgrößen verschiedenster Datentypen, die von den Hard- oder Softwareschnittstellen in die Datenbasis überführt werden. Tabelle 3 zeigt mögliche Datenquellen mit beispielhaften Werten zur Bestimmung des Prüflingsstatus *Türöffnung*.

Datenquelle	Datentyp	mögliche Zustände
Näherungssensor	Boolean	Kontakt <i>Tür geschlossen</i> kein Kontakt <i>Tür offen</i>
Drehwinkelgeber	Gleitkommazahl 64 Bit, e(11), m(52)	<1° <i>Tür geschlossen</i> ≥1° <i>Tür offen</i>
Fahrzeugbus	Byte	0xFC <i>Tür geschlossen</i> 0xFD <i>Tür offen</i> 0xFE <i>Signal unplausibel</i> 0xFF <i>Signal ungültig</i>

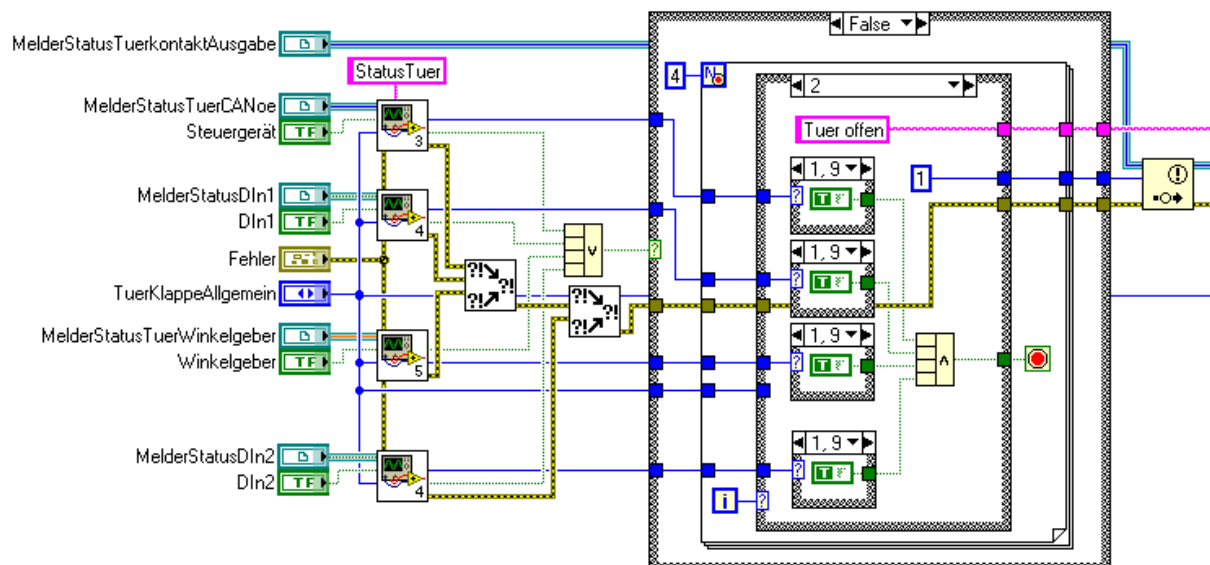
Tabelle 3: Datenquellen zur Bestimmung des Prüflingsstatus *Türöffnung*

Über die Bedienoberfläche der Komponente *Auswertungskonfigurator* lassen sich die benötigten Datenquellen in Abhängigkeit von Anforderung und Verfügbarkeit beim jeweiligen Versuchsaufbau zur Bestimmung des Status *Türöffnung* aktivieren.



**Abbildung 32:** Bedienoberfläche zur Auswahl der Datenquellen zur Bestimmung des Prüflingsstatus *Türöffnung*

Ist der fahrzeugeigene Türkontaktsensor hinreichend abgesichert und im Rahmen der anstehenden Erprobung nicht prüfungsrelevant, reicht der komfortabel auszulesende Fahrzeugbus als einzige Datenquelle aus. Mehr Sicherheit bietet die Hinzunahme eines Näherungssensors, welcher aber aufwendig zu installieren ist. Absolute Sicherheit für Freigabeuntersuchungen von Neuentwicklungen ist durch die Verknüpfung aller drei gelisteten unabhängigen Datenquellen gegeben.



**Abbildung 33:** Programmcode zur Berechnung des Prüflingsstatus *Türöffnung*

Abbildung 33 beinhaltet den Programmcode zur Bestimmung des Prüflingsstatus. Links sind zunächst die Eingangssignale mit ihren verschiedenen Datentypen positioniert. Diese werden im ersten Verarbeitungsschritt zur Vergleichbarkeit unter Berücksichtigung eingestellter Parameter in einen dem Ausgabewert entsprechenden Datentyp konvertiert. Durch die bedingte *For – Schleife* erfolgt im zweiten Schritt die Überprüfung der Eingangswerte auf Übereinstimmungen mit definierten Mustern. Bei einer Übereinstimmung wird der damit verbundene Zustand als aktualisierter Prüflingsstatus ausgegeben. Ist die Kombination der aktiven Eingangssignale nicht definiert oder gegensätzlich, ist der Prüflingsstatus *undefiniert*. Die Aktualisierungsrate hängt von der Taktung der Datenquellen ab und beträgt im Optimalfall 1.000 Hertz.



Der interne Aufbau dieser Komponente ist soweit wie möglich modular. Allerdings sind für jeden einzelnen Prüflingsstatus eigene Datenkonvertierungen und Vergleichsoperationen notwendig. Weitere Beispiele sind *Status Antrieb Fensterheber*, *Position Schiebedach* oder *Bordnetzspannung*. Eine universelle Bedienoberfläche für alle erdenklichen Datenquellen wäre hinsichtlich der Bedienung sowie bei der Erstellung des Programmcodes zu komplex und umfangreich. Erweiterungen dieser Komponente können nur durch Softwareexperten erfolgen. Der Bediener kann lediglich Änderungen den Einstellungen vornehmen.

#### 4.3.5 Ansteuerungskonfigurator

Das Gegenstück zum *Auswertungskonfigurator* ist die Komponente *Ansteuerungskonfigurator*. Hier erfolgt die Konkretisierung von abstrakten Ansteueranweisungen durch eine Zuordnung von Methoden oder Aktoren. Die Komponente garantiert gleichermaßen eine Flexibilität in Bezug auf Prüflingsausstattung, erforderliche Prüftiefe und auftragsspezifische Anforderungen. Die Notwendigkeit dieser Komponente lässt sich beispielhaft anhand verschiedener Varianten zur Betätigung eines Fensterhebers belegen.

Level	Ansteuervariante	Hard- / Software
1	Betätigung Schalterblock Tür	Roboter
2	Kurzschluss der Kontakte Schalterblock Tür	Relais
3	Softwaresimulation Schalterblock Tür	CANoe LIN – Bus
4	Softwaresimulation Komfort – Fensterheberansteuerung	CANoe CAN – Bus
5	Diagnosebefehl an Steuergerät	CANoe CAN – Bus
6	Direktbestromung Fensterhebermotor	Relais

**Tabelle 4: Varianten zur Ansteuerung des Fensterhebers**

Tabelle 4 verdeutlicht, dass mit dem Anstieg des Levels die Prüftiefe abnimmt, weil immer weniger Bestandteile des Gesamtsystems *Fensterheber* in die Prüfung involviert sind. Level 1 ist von der Art der Ausführung absolut kundennah, birgt aber wesentlich mehr Fehlerquellen in der Prüfsystemautomatisierung, blockiert den Roboter für anderweitige Aufgaben, bedarf einer längeren Inbetriebnahme und ist nur bei vollständiger Teileverfügbarkeit des Prüflings durchführbar. Level 6 ist in keiner Form mehr realitätsnah, lässt sich dafür in wenigen Minuten in Betrieb nehmen und reicht für bestimmte Aufgabenstellungen, beispielsweise zur Ermittlung des Dichtungsabriebs, aus.

Komponentenintern erfolgt eine Umwandlung der von den Prüfmodulen generierten aufgabenorientierten Anweisungen in Zustände oder Befehle, die von den ausgewählten Hard- oder Softwarekomponenten interpretierbar sind. Als Beispiel wird der Befehl *Fenster öffnen* bei der Variante *Direktbestromung Fensterhebermotor* in die entsprechenden Schaltzustände der ausgewählten Relais umgewandelt. Bei der Variante *Simulation Schalterblock Tür* ändern sich Umgebungsvariablen in CANoe und damit die zu sendenden Daten auf dem LIN – Bus. Die Einstellungen zum jeweiligen Prüfauftrag können aufgrund der angepassten Oberfläche seitens der Bediener intuitiv konfiguriert werden, interne Änderungen und Ergänzungen sind Softwareexperten vorbehalten. Ebenso wie beim Auswertungskonfigurator erweitert jede zusätzliche Ansteuervariante den Funktionsumfang und die Flexibilität des Prüfsystems.

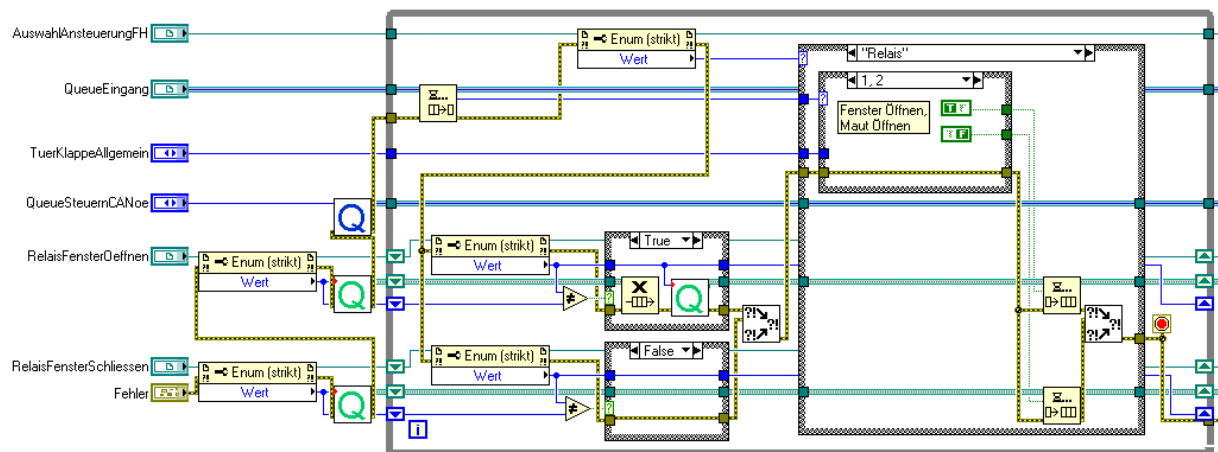


Abbildung 34: Programmcode der Ansteuervariante *Direktbestromung Fensterhebermotor*

Abbildung 34 stellt den Programmcode in LabVIEW dar. Zu Beginn der *While – Schleife* pausiert die Datenverarbeitung, bis über die Queue *Eingang* eine Ansteueranweisung eintrifft. In Abhängigkeit der ausgewählten Ansteuervariante werden in der *Case – Struktur* die Anweisungen in Zustandsänderungen in den jeweils passenden Datentyp umgewandelt und über die *Datenbasis* an die zuständigen Hard- oder Softwareschnittstellen übergeben.

Erfahrungen aus der Praxis zeigen, dass jeder Prüfling einen individuellen Ausstattungs- und Reifegrad besitzt. Der genaue Zustand und die dadurch möglichen Ansteuervarianten sind häufig erst wenige Tage vor Inbetriebnahme bekannt und zudem vom Vorverwender abhängig. Treten während der Lebensdauererprobung Defekte auf, kann bei fehlenden Ersatzteilen eine kurzfristige Veränderung der Betätigungsvariante nötig sein. Die Komponente Ansteuerungskonfigurator hat einen gewichtigen Anteil an der Erfüllung der Anforderung einer zeitgerafften Erprobung, da diese eine schnelle und flexible Inbetriebnahme ermöglicht und bei Defekten lange Prüfungsunterbrechungen durch eine sofortige Umschaltung auf alternative Ansteuervarianten verhindert.

#### 4.3.6 Messdatenverwaltung

Der Wunsch nach einem Prüfsystemkonzept mit einer umfangreichen und kontinuierlichen Messdatenerfassung in Verbindung mit den Methoden aus Abschnitt 4.4 führt zu einem hohen Datenaufkommen, welches eine strukturierte und zweckmäßige Verwaltung erfordert. Die Aufnahme und Verarbeitung der Messdaten erfolgt durch die einzelnen Prüfmodule. Dabei ist zwischen einer *Messkurve* und einem *charakteristischen Wert* zu unterscheiden. Messkurven bestehen im Prüfsystem grundsätzlich aus über der Zeit aufgenommenen Messwerten, wie Stromfluss oder Bordnetzspannung bei der Ansteuerung eines Fensterheberrmotors. Aus diesen Daten lässt sich anschließend ein charakteristischer Wert, in diesem Fall die Betätigungsenergie, berechnen. Dessen Darstellung erfolgt über dem Zyklus, um die Veränderungen über die gesamte Erprobungsdauer zu beschreiben.

Sämtliche Messkurven eines Zyklus werden nach Bauteilen, beispielsweise Tür VL, Schiebedach oder Heckklappe, getrennt und mit zusätzlichen Informationen, wie Start-, Endzeitpunkt sowie Abtastrate, in einer gemeinsamen Datei abgelegt. Die Speicherung von charakteristischen Werten erfolgt mit dem dazugehörigen Zyklus ebenfalls nach Bauteilen getrennt. Diese Struktur ermöglicht einen kurzfristigen Zugriff auf die Daten, da keine große Datenmenge zu durchsuchen ist.

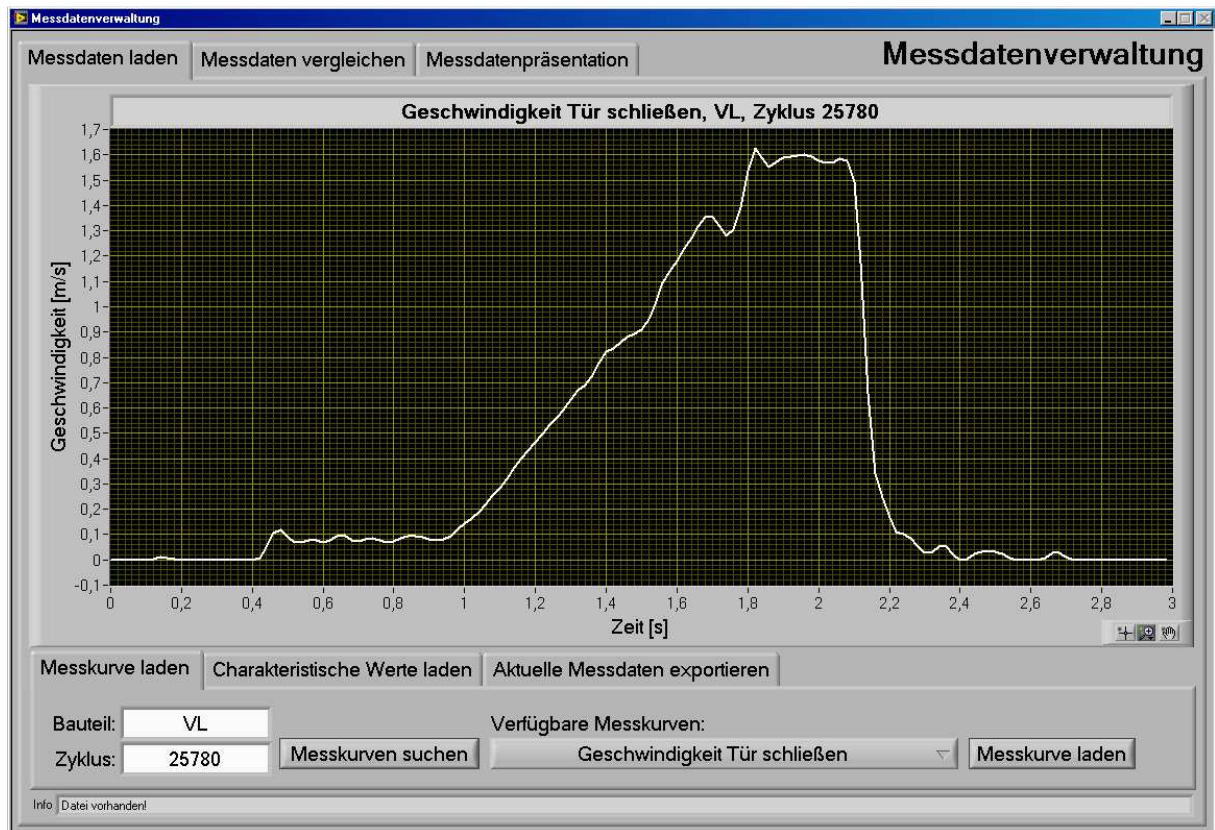


Abbildung 35: Bedienoberfläche der Komponente *Messdatenverwaltung*

Für eine Weiterverarbeitung der Messdaten steht eine Exportfunktion zur Verfügung, welche die aktuell angezeigten Messdaten in einer Textdatei im ASCII – Format mit Zeit- oder Zyklusvektor speichert. Die Komponente *Messdatenverwaltung* verfügt über keine speziellen Funktionen zur Weiterverarbeitung, wie Filter oder Statistikfunktionen. Dafür bieten sich mit beispielsweise *DIAdem (NI)* oder *Excel (Microsoft)* diverse Softwarewerkzeuge von Drittanbietern an, in welche sich die Messdaten importieren lassen.

Zur Anzeige von Messdaten stehen dem Bediener drei Reiter zur Auswahl. Der Reiter *Messdaten laden* dient zur Selektion und Anzeige einer einzelnen Messkurve oder Sammlung von charakteristischen Werten. Die geladenen Messdaten können zum direkten Vergleich mit anderen Kurven in einen von vier Graphen im zweiten Reiter *Messdaten vergleichen* kopiert werden. Das dritte Fenster beinhaltet die Konfiguration und Anzeige der *Messdatenpräsentation*, eine detaillierte Beschreibung dieser Funktionalität erfolgt in Abschnitt 4.4.

#### 4.3.7 Mess- und Analysetool

Mit der Komponente *Mess- und Analysetool* kann jederzeit auf die Melder der Kategorien *Hard- und Softwarestatus*, *Prüflingsstatus* sowie *Ansteueranweisungen* innerhalb der Datenbasis zugegriffen werden. Dadurch sind die dynamischen internen Zustände des Prüfsystems über die Zeit erfassbar, was die Möglichkeiten der Messdatenerfassung und Analyse beträchtlich erweitert. Das Tool kann wahlweise zu den für die aktuelle Prüfung relevanten Messdaten auch weitere Hardwarekanäle oder spezielle Umgebungsvariablen aus CANoe abfragen. Dies ist nützlich, weil einige Messmittel mangels Klimatauglichkeit nicht in die Automatisierung des Prüfablaufs eingebunden sind, die Dauer einer automatischen Messung im speziellen Untersuchungsfall zu gering ist oder bei einer Bauteilanalyse eine manuelle Zusatzmessung er-



forderlich ist. Dabei ist es unerheblich, ob diese Messung zeitgleich mit einem Prüfablauf oder unabhängig davon stattfindet.

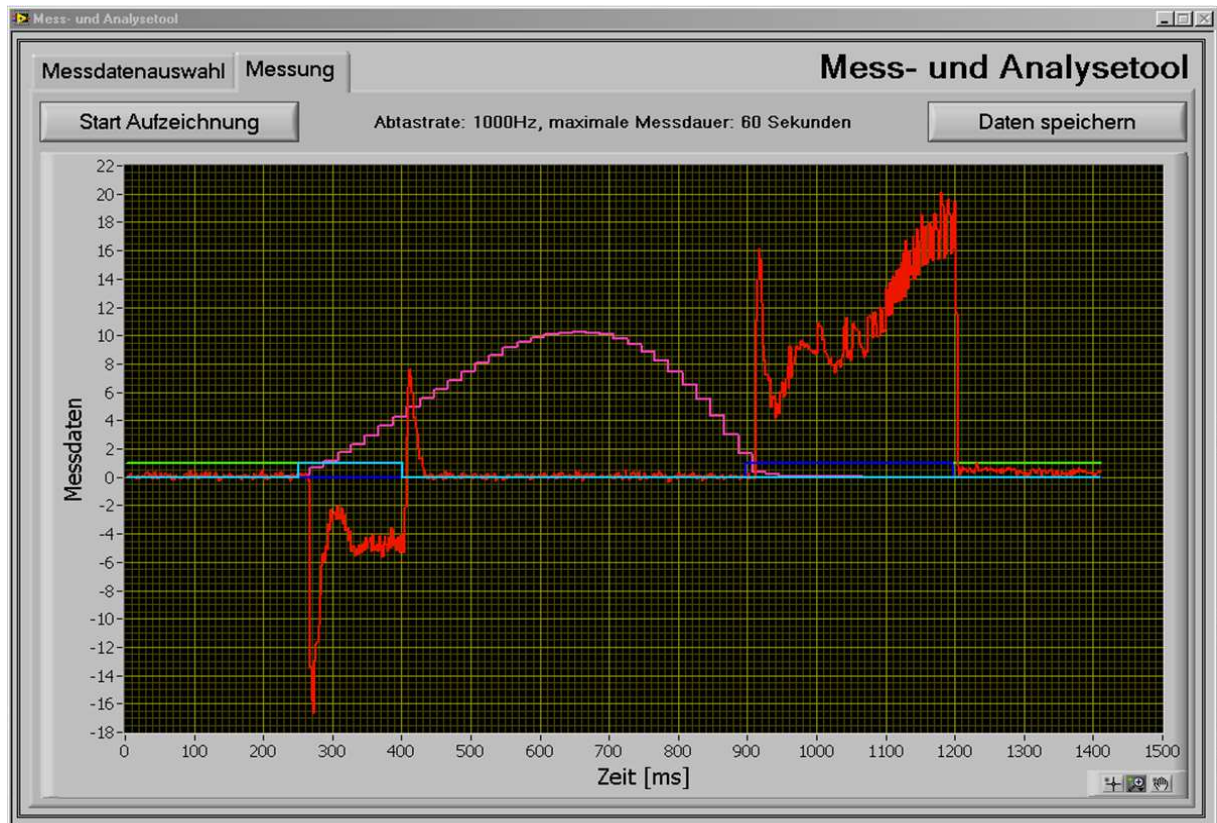


Abbildung 36: Bedienoberfläche der Komponente *Mess- und Analysetool*

Bei der Inbetriebnahme eines Prüfaufbaus können mit Hilfe dieses Tools Parameter und Sensoren präziser und in kürzerer Zeit eingestellt werden. Abbildung 36 beschreibt ein Beispiel zur Parametrierung der Kurzhub – Funktion durch Direktbestromung des Fensterhebermotors samt Ermittlung des Status *Türöffnung* über einen Näherungssensor. Die rote Kurve zeigt die Stromaufnahme des Fensterhebermotors, der Türöffnungswinkel ist violett dargestellt. Mittels der grünen Datenreihe wird der Status *Türöffnung* abgebildet, die blauen Kurven visualisieren die Schaltzeiten der Relais. Aus diesen Messdaten lässt sich erkennen, ob der Abstand des Näherungssensors zur Tür passend eingestellt ist und inwieweit sich der Status *Türöffnung* beim eingestellten Türöffnungswinkel verändert. Weiterhin ist direkt deutlich, ob die Bestromungszeit des Motors für den Kurzhub und die daran anschließende Blockfahrt den Herstellervorgaben entspricht. Ohne dieses Tool sind entweder aufwendige Messaufbauten erforderlich oder es ist im Gegensatz zur millisekundengenauen Einstellbarkeit lediglich eine grobe Abschätzung der Parameter möglich.

Prinzipiell können mit dem Mess- und Analysetool bei ausreichender Rechnerkapazität sämtliche in Abbildung 25 dargestellten Melder aus der Datenbasis in die Messung einbezogen werden. Aus Gründen der Übersichtlichkeit sind selten mehr als fünf Messkanäle sinnvoll. Jeder Messgröße kann eine individuelle Farbe zugeordnet werden. Die Kurve wird online parallel zur Messung aktualisiert und anschließend in einer ASCII – Datei zur weiteren Verwendung gespeichert.

Eine hinreichend genaue Synchronisation der Messkanäle wird erreicht, da die Hard- und Softwareschnittstellen die eingelesenen Daten ohne Zeitversatz in die Datenbasis überführen. Eine maximale Systemauslastung des Prüfstandsrechners von acht Prozent während der parallelen Ausführung sämtlicher Komponenten des Prüfsystems lässt den Schluss zu, dass die

Messung im Rahmen der *weichen Echtzeit* zuverlässig abläuft. Messdaten, die aus genannten Gründen nicht mit einer Rate von 1.000 Hertz aktualisiert werden, sind in der Messkurve bei entsprechend kleiner Auflösung als Treppensignal erkennbar. Der Zeitvektor wird bei jeder Messung aus dem System – Timer generiert. Dies ermöglicht eine weitere Kontrolle der Messgenauigkeit. Verzögerungen wären direkt durch Lücken in der Beschriftung der Abszisse im Diagramm erkennbar.

#### 4.3.8 Ereignisanzeige

Die *Ereignisanzeige* ist im Prüfsystemkonzept die zentrale Komponente zur Anzeige, Speicherung und Kommunikation von Informationen. Dabei wird zwischen *Dauerlauf – Dokumentation* und *System – Dokumentation* unterschieden. Ein Bestandteil der Dauerlauf – Dokumentation sind Fehler am Prüfling, die von einmaligen Funktionsstörungen ohne weitere Auswirkungen bis zu kritischen Fehlern mit gleichzeitiger Unterbrechung der Erprobung reichen. Der andere Teil beinhaltet eine chronologische Auflistung von Zählerständen der Prüfzyklen, aktivierte Prüfmodule sowie Laufzeiten für eine Nachkalkulation. Anhand dieser Daten lässt sich jederzeit der gesamte Prüfablauf samt Fehlfunktionen rekonstruieren, damit ist die Forderung nach einer lückenlosen Protokollierung der Ereignisse umgesetzt.

Der Reiter zur Darstellung der Systemfehler dient zur Überwachung der Systemstabilität durch den Softwareexperten. Hier werden Fehler im Zusammenhang mit der Ausführung von internen Funktionen, während der Kommunikation mit Hard- oder Softwareschnittstellen, beim Lesen oder Speichern von Parametern oder bei Messbereichsüberschreitungen der externen Sensorik registriert. Timeout – Fehler, die bei einer Zeitverzögerung in der Datenanlieferung zum Auswertungskonfigurator gehäuft auftreten, sind ein Indikator für mangelnde weiche Echtzeitfähigkeit und verlangen nach Optimierungen in der Prüfsystemsoftware oder Rechnerhardware. Die Speicherung der Dauerlauf- und Systemfehler – Dokumentation erfolgt in getrennten Dateien, welche komfortabel und gezielt mit Hilfe der *AutoFilter – Funktion* in *Excel* ausgewertet werden können.



Abbildung 37: Bedienoberfläche der Komponente *Ereignisanzeige* und GSM – Mobiltelefon

Über die Komponente Ereignisanzeige erfolgt die Benachrichtigung der betreuenden Versuchingenieure bei einer unplanmäßigen Unterbrechung der Prüfung. Die zunächst ange-

wandte Variante als Email – Benachrichtigung zeigt zwei Nachteile und ist daher nicht optimal. Erstens muss der Prüfstandsrechner dafür im Firmennetzwerk integriert sein, was Performanceeinbußen durch Virenschutzprogramme und weitere sicherheitsbedingte Einschränkungen bedeutet. Zweitens ist zur effektiven Vermeidung von Stillständen ein permanenter Zugang der Betreuer zum Email – Postfach nötig, was im Versuchswesen durch häufige Werkstattarbeiten praktisch nicht realisierbar ist.

Die alternative Benachrichtigung per SMS hat sich in der Praxis bewährt. Die betreuenden Mitarbeiter können ihrer Arbeit nachgehen, ohne regelmäßig einen Blick auf den Prüfablauf richten zu müssen. Beim Vorliegen einer Störung erfolgt unverzüglich die gewünschte Übermittlung an die Einträge der SMS – Empfangsliste. Die Überwachung eines Dauerlaufs außerhalb der betrieblichen Arbeitszeit ist für die Anforderung der zeitgerafften Erprobung von besonderer Bedeutung, da Dauerversuche im Optimalfall zwei Drittel außerhalb dieser Zeit ablaufen. Ein Notdienst kann bei Störungen reagieren, aufgetretene Fehler beheben und die Prüfung wieder starten. Ebenfalls als sehr nützlich hat sich erwiesen, Mitarbeitern im SMS – Verteiler kurze Nachrichten, beispielsweise über die Ursache eines gerade behobenen Fehlers, zukommen zu lassen. So ist jeder Betreuer über den aktuellen Stand informiert, auch wenn er sich gerade nicht vor Ort aufhält.

#### 4.3.9 Prüfablaufverwaltung

Die Komponente *Prüfablaufverwaltung* gliedert sich in einen *Prüfplaneditor* zur individuellen Gestaltung von Prüfabfolgen sowie in die *Prüfablaufsteuerung* zum Starten, Überwachen und Stoppen der Erprobung.

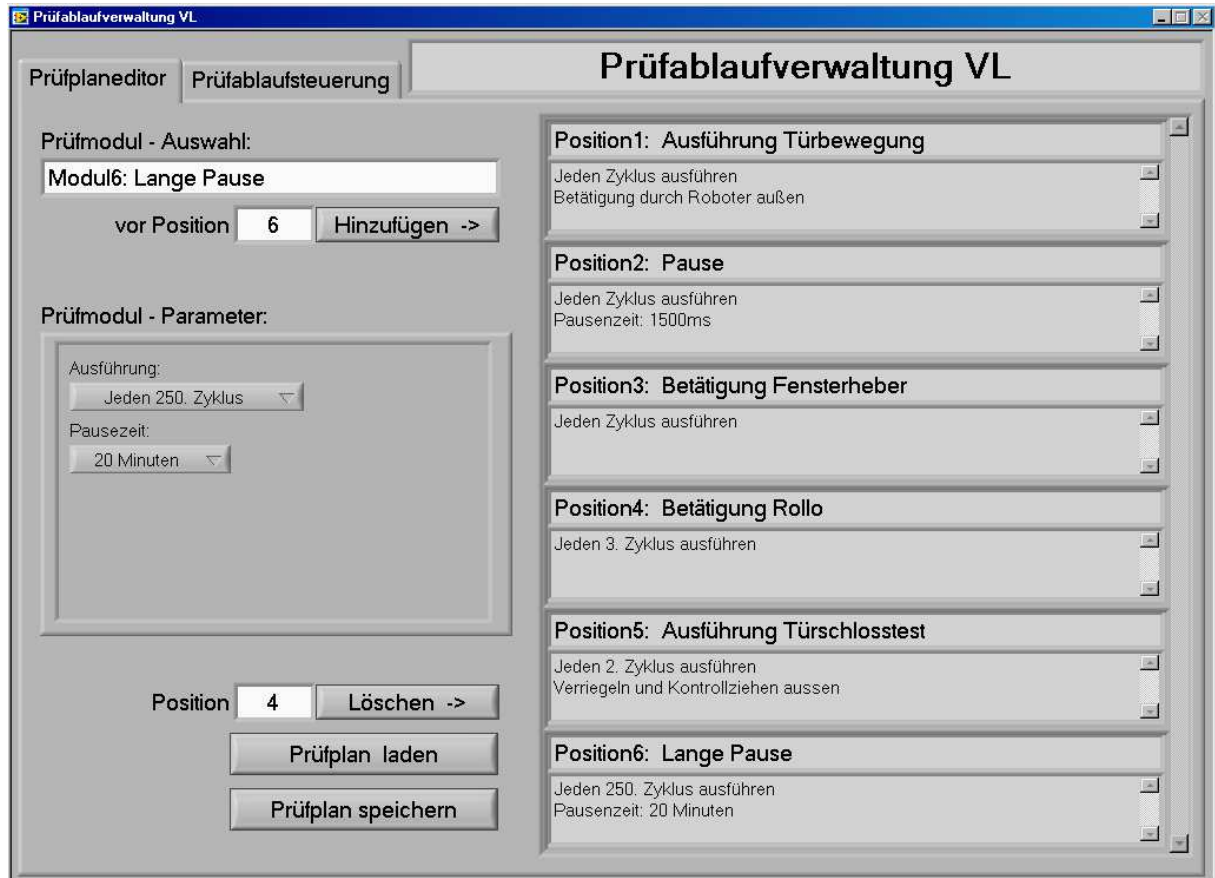


Abbildung 38: *Prüfplaneditor* der Komponente *Prüfablaufverwaltung*



Im Prüfplaneditor stehen die Prüfmodule zur Auswahl, die durch Softwareexperten bereits ins Prüfsystem implementiert sind. Beim Hinzufügen eines Prüfmoduls in den Prüfplan sind modulbezogene Parameter einzustellen, zum Beispiel die Variante *Roboter innen* oder *Roboter außen* bei einer Türbewegung. Diese *Prüfmodulparameter* stehen im direkten Zusammenhang mit dem Prüfmodul und dessen Position im Prüfplan und können nicht allgemein in der Komponente *Parameterverwaltung* hinterlegt werden. Erstellte Prüfablaufpläne lassen sich zur erneuten Verwendung oder als Vorlage speichern und wieder laden.

Die Prüfablaufsteuerung dient der automatisierten Ausführung der im Prüfplan enthaltenen Prüfmodule in Form eines Dauerlaufs. Dazu gehören das Starten und Stoppen des Prüfablaufs, die Einstellung von Zielzyklen und eine Anzeige der aktuell ausgeführten Vorgänge am Prüfling. Währenddessen speichert die Prüfablaufsteuerung Umweltbedingungen, wie Bordnetzspannung und Umgebungstemperatur sowie Fehleranalyse und Reproduzierbarkeit für die spätere Berichterstellung.

Ein weiterer Bestandteil ist die Berechnung der durchschnittlichen Laufzeit eines Prüfzyklus. Auf Basis dieser Zeiten lassen sich deutlich optimierte Auslastungsplanungen für Prüfmittel erstellen, und die Versuchingenieure können Analyse- und Umbauzeiten wesentlich effektiver in den Betriebsablauf einplanen. Das prozentuale Verhältnis von Laufzeit und ungeplanten Prüfungsunterbrechungen ist ein Indikator für die Qualität von Prüfling, Prüfstand und Betreuung. Bei Werten unterhalb von 70 Prozent ist auf den ersten Blick erkennbar, dass die kalkulierte Durchlaufzeit gefährdet ist. Besonders bei zeitkritischen Prüfungen wäre eine Intensivierung der Betreuung erforderlich.

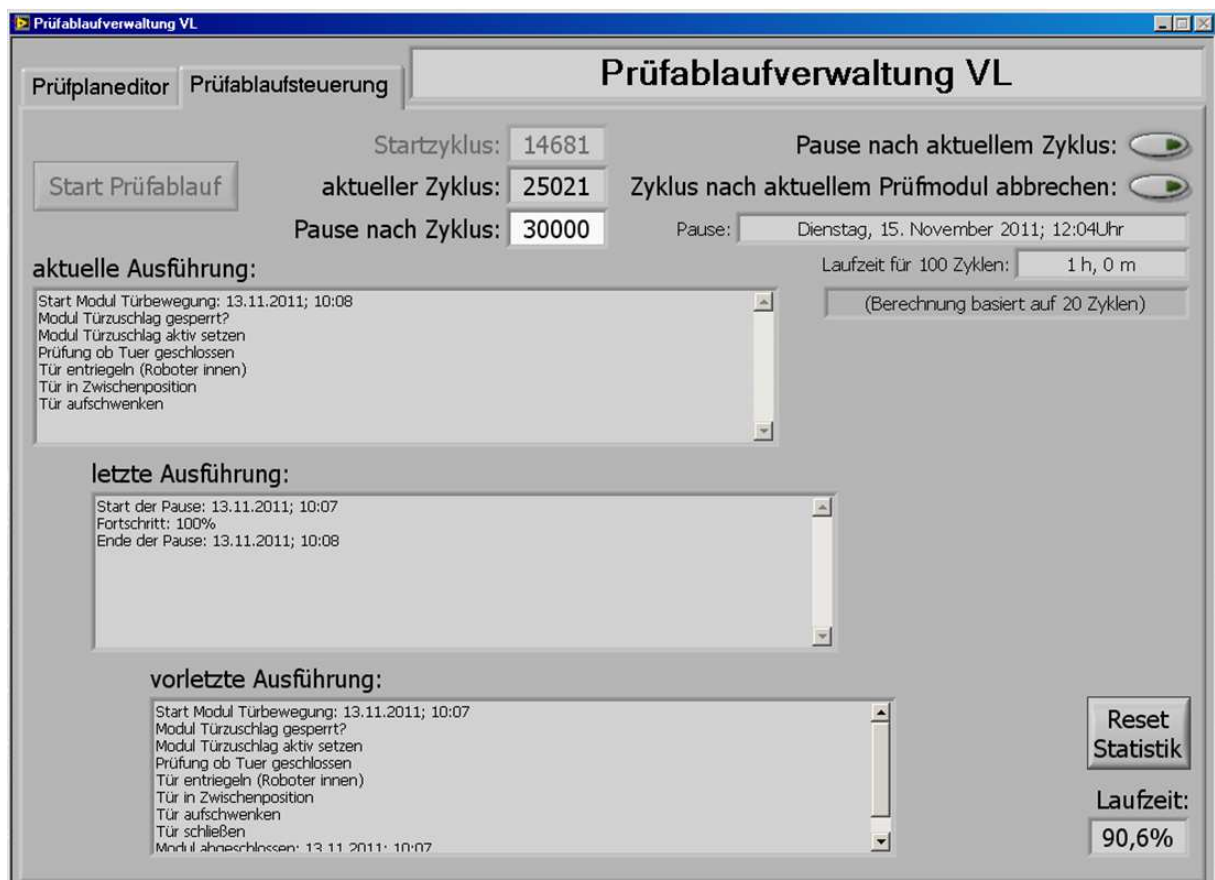


Abbildung 39: Prüfablaufsteuerung der Komponente Prüfablaufverwaltung

#### 4.3.10 Prüfmodule

*Prüfmodule* sind die Komponenten des Prüfsystemkonzepts, welche die individuellen Eigenheiten der jeweiligen Prüfungen im Zusammenhang mit dem zu erprobenden Bauteil berücksichtigen. Der softwaretechnische Aufbau ist soweit wie möglich aus universell verwendbaren *Basisprogrammen* zusammengesetzt. Zu den Aufgaben dieser Basisprogramme gehören die Übertragung von Ansteuerbefehlen an die Datenbasis, die Abfrage des Prüflingsstatus sowie der Parameter oder die Erfassung, Verarbeitung und Weiterleitung von Messdaten.

Der interne Ablauf der Prüfmodule ist durch einen *virtuellen Zustandsautomaten* geprägt, welcher die parallelen und sequentiellen Aufrufe der Basisprogramme sowie des prüfungs- und bauteilspezifischen Programmcodes organisiert. Zustandsautomaten bieten sich in Verbindung mit Variablen des Typs *enum* bei grafischen Programmiersprachen als eine sehr übersichtliche Methode an, um den Aufwand von Änderungen und Ergänzungen zu minimieren. Dadurch ist der interne Ablauf eines Prüfmoduls für Softwareexperten ohne Bezug zur konkreten Erprobungsaufgabe unschwer nachvollziehbar, wodurch eine wesentliche Voraussetzung zur kurzfristigen Implementierung neuer Prüfmodule erfüllt ist. Ein weiterer Vorteil der Zustandsautomaten, nämlich aus jedem Zustand heraus in jeden beliebigen Zustand wechseln zu können, kommt meistens nicht zum Tragen, da der Ablauf eines Prüfmoduls grundsätzlich fest definiert ist. Im Fehlerfall ist dadurch eine unverzügliche Beendigung des Prüfmoduls möglich.

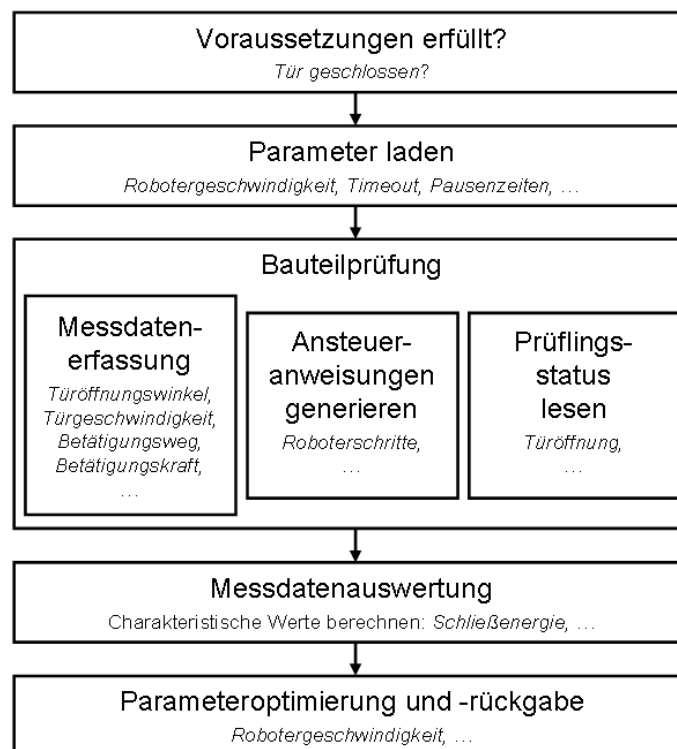


Abbildung 40: Interner Aufbau der Prüfmodule

Die elementare Reihenfolge der einzelnen Zustandsdiagramme ist abgesehen von prüfungstypischen Eigenheiten bei jedem Prüfmodul identisch. Zu Beginn findet eine Überprüfung der Voraussetzungen für die Durchführung des Prüfmoduls statt. Im positiven Fall werden die zur Ausführung benötigten Parameter aus der Datenbasis gelesen. Die Ausführung der eigentlichen Prüfung beginnt mit dem Start der Messdatenaufzeichnung, wenige Millisekunden verzögert setzen parallel die Ansteuerung des Prüflings sowie die Überwachung seines Status ein. Ist die gewünschte Änderung am Prüflingsstatus eingetreten, endet die Messdatenauf-



zeichnung. Im Anschluss erfolgen die Auswertung der Aufzeichnung sowie mögliche Rückgaben geänderter Parameter an die Datenbasis. Sind im Verlauf der Abarbeitung des Prüfmoduls Fehlfunktionen oder Zeitüberschreitungen aufgetreten, werden diese an die Ereignisanzeige weitergeleitet.

Die Ausführung eines Prüfmoduls soll weitestgehend unabhängig von parallel laufenden Prüfungen stattfinden. Für eine Türerprobung an einem viertürigen Fahrzeug bedeutet dies, dass jede Tür unabhängig und im individuellen Tempo den Prüfplan durchläuft. Eine konsequente Einhaltung ist dann nicht möglich, wenn beispielsweise Diagnosefunktionen bestimmte Prüfungsstatus voraussetzen oder Steuergeräte nur eine gemeinsame Ansteuerung mehrerer Antriebe vorsehen. So lassen sich der linke und rechte Außenspiegel nur zusammen aus- oder beklappen. Für diese Abhängigkeiten sind in der Datenbasis *Merker* vorhanden, welche einen Datenaustausch zwischen den Prüfmodulen ermöglichen.

#### 4.3.11 Softwaretools für Einzelprüfungen

Neben Dauerversuchen finden während der Komponentenerprobung auch Einzelversuche statt. Beispiele sind die Messung von Bauteilsteifigkeiten, die Betätigungskräfte zum Schließen einer Klappe oder die Aufzeichnung des Temperaturverlaufs einer Sitzheizung. Ein mittelfristiges Ziel dieser Arbeit ist es, eine Auswahl an Softwaretools zu implementieren, mit deren Hilfe diese überwiegend halbautomatischen Einzelversuche effektiver ausführbar sind. Die softwarebasierte Systemarchitektur, insbesondere die Hard- und Softwareschnittstellen, das Mess- und Analysetool sowie der uneingeschränkte Zugriff auf die Datenbasis, stellen dafür eine optimale Grundlage dar. Bei der Erstellung der Softwaretools können die Softwareexperten zudem auf bereits erprobte Basisprogramme zurückgreifen.

Da die Prüfsystemsoftware auf jedem LabVIEW – fähigen Rechner lauffähig ist und eine hohe Flexibilität beim Anschluss von Datenerfassungshardware aufweist, sind die Softwaretools in der Praxis in maximal einer Stunde einsatzbereit. Die Bedienoberfläche weist dabei viele Parallelen zum übrigen Erscheinungsbild der Prüfsystemsoftware auf, der Einarbeitungsaufwand ist daher gering. Als Beispiel wird in Kapitel 6 der Aufbau des Softwaretools *Steifigkeitsmessung* beschrieben.

#### 4.3.12 Hauptmenü

Das Hauptmenü der Prüfsystemsoftware *DaLaBert* hat die Aufgabe, einen komfortablen Zugang zu sämtlichen geladenen Komponenten zu ermöglichen. Ist die Bedienung der Oberfläche nicht durch die Eingabe des korrekten Kennworts freigegeben, sind gemäß den Anforderungen alle Bedienelemente gegen unberechtigten Zugriff gesperrt. Lediglich die aktuellen Zyklenstände der Prüfablaufsteuerung und der farblich gekennzeichnete Status sind sichtbar. Grün bedeutet aktiv, grau kennzeichnet eine Pause und der rote Hintergrund deutet auf eine Störung hin.

Abbildung 41: Bedienoberfläche des *DaLaBert* Hauptmenüs

## 4.4 Messdatenerfassung und Analyse

Die Anforderungen nach einer zeitgerafften und realitätsnahen Lebensdauererprobung sowie einer umfangreichen Analysefunktionalität verbindet die Notwendigkeit einer ausgeweiteten Messdatenerfassung mit adäquaten Auswertungs- und Exportfunktionen. Der Raffungsfaktor einer Prüfung kann durch die Automatisierung vormals manueller Messungen gesteigert werden. Die geforderte Realitätsnähe eines Versuchs lässt sich detaillierter überwachen und die Fehleranalyse ist umso aussagekräftiger, je umfangreicher Messwerte zur Verfügung stehen.

Beim Betrieb technischer Anlagen besteht oftmals die Problematik, dass die große Fülle von Messdaten aus Zeit- und Kostengründen nicht oder nur unzureichend ausgewertet werden kann [Wie-05]. Die folgenden Kapitel beschreiben eine halbautomatische Lösung, welche den Versuchingenieuren umfassend und schnell eine Übersicht über den Prüfablauf bietet.

### 4.4.1 Kontinuierliche Messdatenerfassung

Das neue Prüfsystemkonzept ist hard- und softwaretechnisch dafür ausgelegt, alle relevanten und mit vertretbarem Aufwand erfassbaren physikalischen Größen im Zusammenhang mit einer Prüfung kontinuierlich aufzuzeichnen. Ein Aufwand gilt als nicht vertretbar, wenn Messstellen nicht oder nur mit größtem Aufwand zugänglich sind, Messmittel zu hohe Kosten verursachen oder Prüfabläufe zu stark beeinträchtigt werden. Manuelle Messungen und damit verbundene Nachteile, beispielsweise ein großer Zeitaufwand, personenbezogene Einflüsse oder diskontinuierliche Messzeitpunkte, entfallen dadurch weitestgehend. Verkürzte Entwicklungszeiten und die Reduzierung der Erprobungsschleifen machen es erforderlich, aus den verbleibenden Tests ein Maximum an Erkenntnissen zu gewinnen.

Für die Erprobung von Karosseriekomponenten bedeutet dies konkret, Spannung und Strom bei der Ansteuerung elektrischer Antriebe sowie Kraft und Weg bei mechanischen Betätigun-

gen zu messen. Bei einer funktionsfähigen Fahrzeugelektronik kommt die Aufzeichnung von Busnachrichten, Fehlerspeichereinträgen oder Steuergerät – Parametern hinzu. Für die Betrachtung und Auswertung der aufgezeichneten Messdaten sind Hilfsmittel notwendig. Eine prinzipielle Vorgehensweise für die Weiterverarbeitung anhand eines *Erkennungssystems* zeigt Abbildung 42.

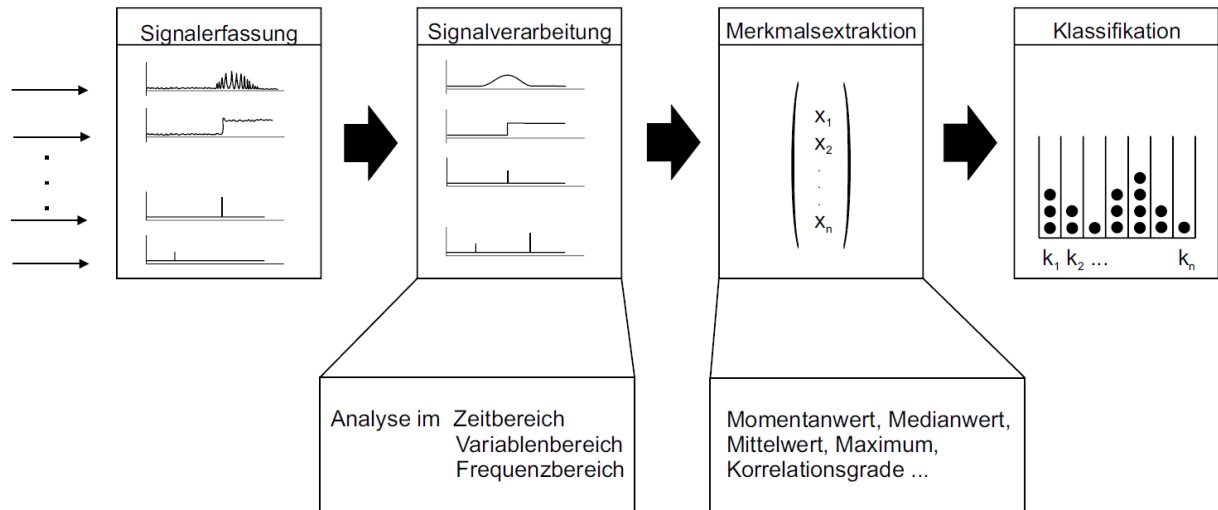


Abbildung 42: Erkennungssystem zur Verarbeitung von Messdaten [Rec-05]

Eine aussagekräftige und zuverlässig funktionierende Klassifikation der gewonnenen Daten ist nur möglich, wenn die charakteristischen Verläufe der zu erwartenden Messkurven im Vorfeld bekannt sind und bei den Methoden zur Extraktion Berücksichtigung finden. Welche Problematik sich daraus für die Anwendung in der Komponentenerprobung abzeichnet, beschreibt Abbildung 43 anhand des nicht repräsentativen Stromverlaufs eines Fensterhebermotors beim Schließvorgang der Scheibe.

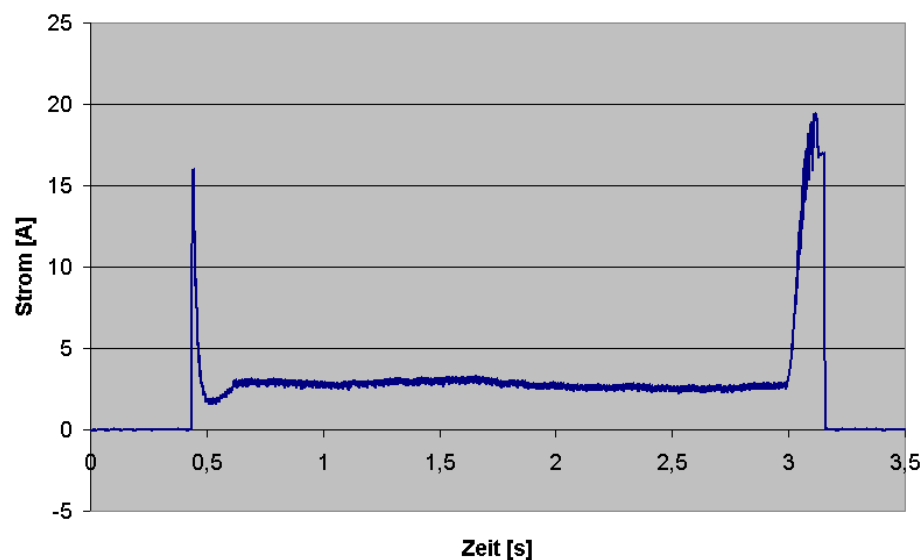


Abbildung 43: Stromverlauf eines Fensterhebermotors

Die erste Spitze ergibt sich primär aus der Trägheit der bewegten Masse sowie Haftreibung zwischen Dichtung und Scheibe beim Beschleunigen der Bewegung. Anschließend setzt die annähernd gleichförmige Scheibenbewegung bei konstantem Stromfluss ein. Der Anstieg zum Ende der Fensterfahrt resultiert aus dem Einfahren der Scheibe in die obere Dichtung sowie der Blockfahrt gegen den Fensterrahmen.

Die Anwendung von Statistikfunktionen, beispielsweise *Mittelwert*, liefert bei der Charakteristik dieses Stromverlaufs keine brauchbaren Ergebnisse. Mit Hilfe von *Korrelationsfunktionen* lässt sich bestimmen, wie groß eine Ähnlichkeit zweier Signalverläufe zueinander ist. Deren Anwendung setzt voraus, dass für jede Messung eine Vergleichskurve im Prüfsystem hinterlegt ist. Diese Bedingung ist bei Komponentenversuchen in der Praxis nicht umsetzbar. Allein für den Fensterheber sind bei einem Dauerlauf bis zu zehn verschiedene Bewegungsformen, wie *Öffnen*, *Schließen* oder *Kurzhub* sowie Kombinationen möglich. Weiterhin ist zu berücksichtigen, dass die Messkurven erfahrungsgemäß während der ersten Zyklen einer Dauererprobung ihre Charakteristik beim sogenannten *Einlaufvorgang* des Bauteils verändern. Erfolgt die Ansteuerung über ein Steuergerät, variiert die Dauer der Blockfahrt durch zyklisch ausgeführte Referenzierungsfunktionen seitens der Elektronik. Zudem ist der starke Einfluss verschiedener klimatischer Umweltbedingungen zu berücksichtigen.

Für eine Bewertung elektrisch angetriebener Systeme hat sich bewährt, über die aufgenommenen Messkurven von Strom und Spannung die Leistung zu berechnen und daraus die *elektrische Betätigungsenergie* zu integrieren. Bei mechanisch betätigten Bauteilen gilt dies entsprechend für die *mechanische Betätigungsenergie*, die sich als Integral der Arbeit über die Messdaten von aufgebrachtener Kraft und zurückgelegtem Weg berechnen lässt. Eine Darstellung der jeweiligen Betätigungsenergie über einen Erprobungszyklus macht Veränderungen an den Prüflingen auf den ersten Blick sichtbar. Bei Messungen mit einem Kraft – Weg – Zusammenhang sind zudem Steifigkeitsberechnungen möglich. Zusätzlich ist es sinnvoll, Maximal- oder Momentanwerte in bestimmten Grenzen zu extrahieren. Dazu gehört beispielsweise die Höchstgeschwindigkeit in einem Winkelfenster beim Türzuschlag oder der Öffnungswinkel einer Tür nach dem Ausschwingen. Schleichende Veränderungen mit zunehmender Zykluszahl beschreiben einen Verschleiß, ein Sprung in der Kurve ist das Resultat eines plötzlichen Bruchs oder Risses. Grundsätzlich ist zu beachten, dass bei sämtlichen Bauteilen für jede Bewegungsform eine separate Auswertung nötig ist. Abbildung 44 zeigt den Verlauf einer geregelten Maximalkraft und die dafür aufzubringende Betätigungsenergie. Die Defekte am Bauteil samt den dazugehörigen Zyklen sind optimal nachvollziehbar, eine Reproduzierbarkeit der Testbedingungen ist gegeben.

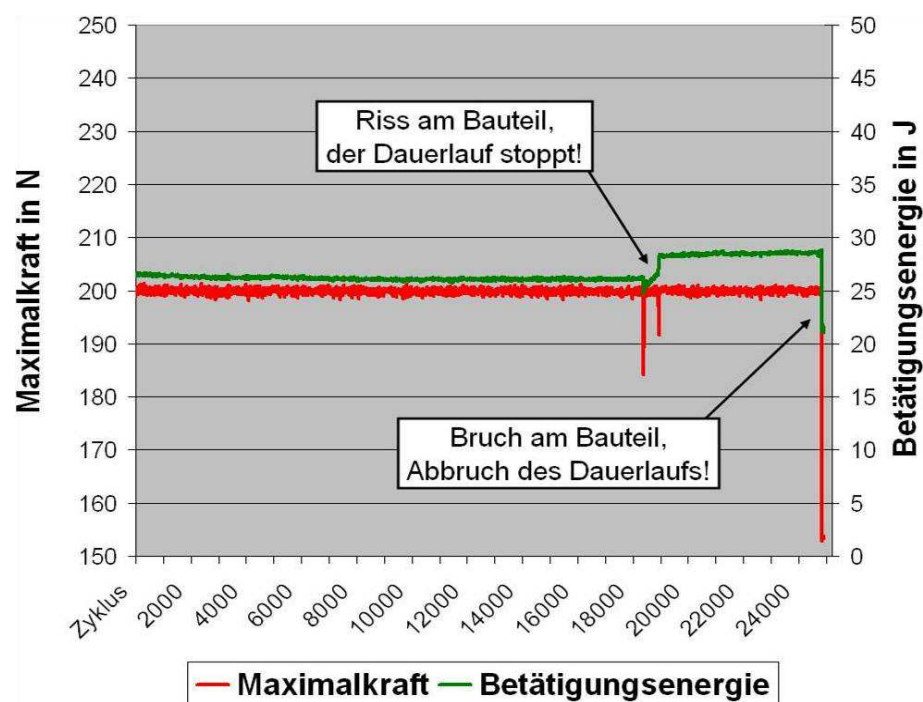


Abbildung 44: Verläufe der Maximalkraft und Betätigungsenergie bei einer Dauererprobung



Trotz der beschriebenen Reduktion von Messdaten durch die Berechnung von charakteristischen Werten entsteht während der Dauererprobung für jedes extrahierte Merkmal eine separate Kurve. Daher ist eine Methode notwendig, welche die für den aktuellen Versuch relevanten Daten aus dieser großen Menge übersichtlich und weitestgehend automatisiert zur Verfügung stellt. Der verantwortliche Versuchsingenieur muss innerhalb weniger Minuten die wichtigsten Informationen über die Güte der zu prüfenden Bauteile, den bisherigen Verlauf sowie den aktuellen Status erhalten. Ausreißer, Sprünge oder Tendenzen in den Kurvenverläufen müssen direkt sichtbar sein.

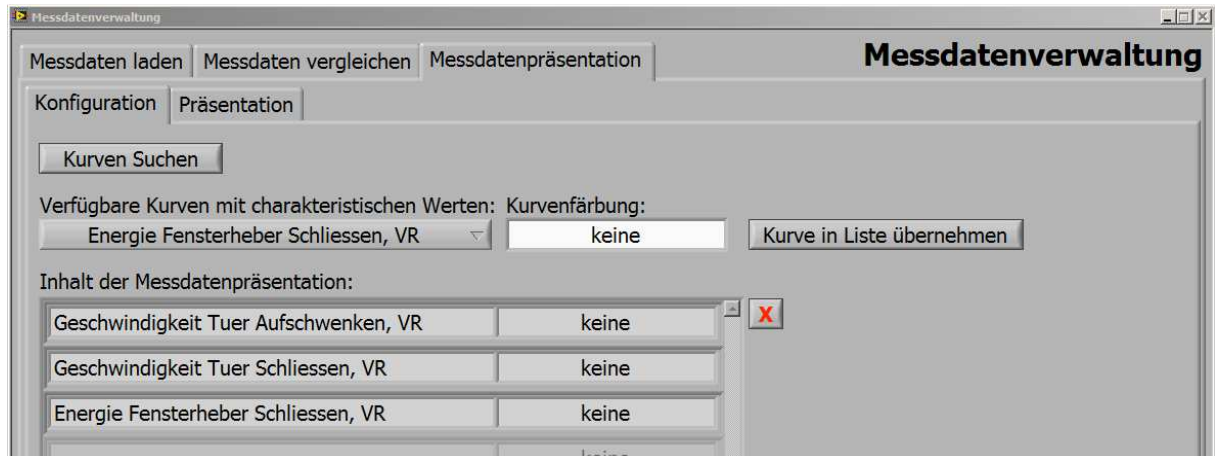


Abbildung 45: Bedienoberfläche zur Konfiguration der Messdatenpräsentation

In der Komponente *Messdatenverwaltung* können verfügbare Messkurven mit charakteristischen Werten in einer halbautomatischen Messdatenpräsentation visualisiert werden. Abbildung 45 zeigt, auf welche Weise der Bediener die relevanten Messkurven in wenigen Schritten zu einer Liste zusammenstellen kann. Im Anschluss erfolgt die Anzeige der ausgewählten Kurven im Stil einer Präsentation, welche über die Schaltfläche *Weiter* gesteuert wird.

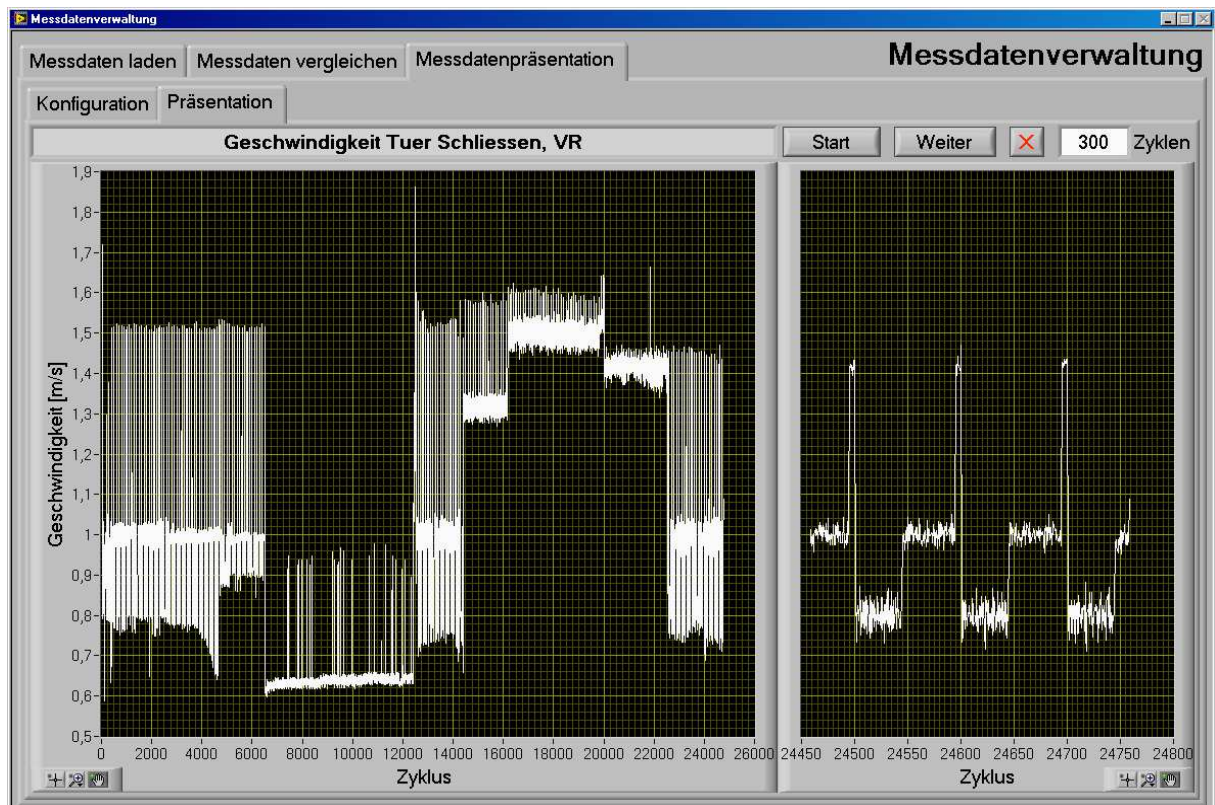


Abbildung 46: Bedienoberfläche zur Anzeige der Messdatenpräsentation

Der linke Graph beschreibt den Verlauf der Schließgeschwindigkeit einer Tür vom Start des Versuchs bis zum aktuellen Zyklus. Im rechten Graphen ist eine einstellbare Anzahl der letzten Zyklen dargestellt. Veränderungen in diesem Prüfabschnitt lassen sich so detailliert erkennen und mit dem gesamten Erprobungsverlauf vergleichen. Durch diese halbautomatische Präsentation ist eine tendenzielle Beurteilung des Gesamtverlaufs inklusive der aktuellen Veränderungen innerhalb weniger Minuten möglich.

Vollständig automatisierte Auswertungsfunktionen, die beispielsweise über ein Ampelsystem die Qualität des Prüflings signalisieren, sind in der Komponentenerprobung nicht praktikabel. Gründe dafür sind die Individualität der jeweiligen Prototypen und die große Anzahl an äußeren Einflussfaktoren. Besonders kritisch ist die Definition und Extraktion von Merkmalen, bei deren Über- oder Unterschreitung die Erprobung automatisch stoppen würde. Diese Methode ist eher geeignet bei *End – of – Line – Tests*, da hierbei große Stückzahlen identischer Bauteile mit eindeutig definierbaren Merkmalen überwacht werden. Bei Komponentenversuchen an Prototypenteilen ist die regelmäßige Überwachung durch erfahrene Mitarbeiter unverzichtbar. So steigt bei hochgradig automatisierten Analysen die Gefahr, dass Störungen vorerst oder grundsätzlich unerkannt bleiben, da der eingetretene Fall nicht vordefiniert ist.

#### 4.4.2 Auswirkung variabler Umgebungsbedingungen

Die Nachstellung der im alltäglichen Gebrauch auf Fahrzeugkomponenten wirkenden Umgebungsbedingungen und die Übertragung der Auswirkungen auf die Realität stellen eine besondere Herausforderung dar. Zu den Umgebungsbedingungen zählen besonders die Temperatur und Luftfeuchte. Für elektronische Systeme ist die anliegende Bordnetzspannung ein wichtiges Kriterium. Softwarefunktionen auf Steuergeräten besitzen eine Abhängigkeit von Inhalten der Busnachrichten *Klemmenstatus*, *Fahrzeuggeschwindigkeit* oder *Karosserieneigungswinkel*.

Die Durchführung einer Dauererprobung mit durchgängig konstanten Umgebungsbedingungen ist nicht realitätsnah. Während des Ablaufs der Lebensdauererprobung müssen möglichst realistische Zustände inklusive deren Schwankungen nachgestellt werden. Abbildung 47 zeigt beispielhaft das Ergebnis einer Spannungsabfallsimulation.

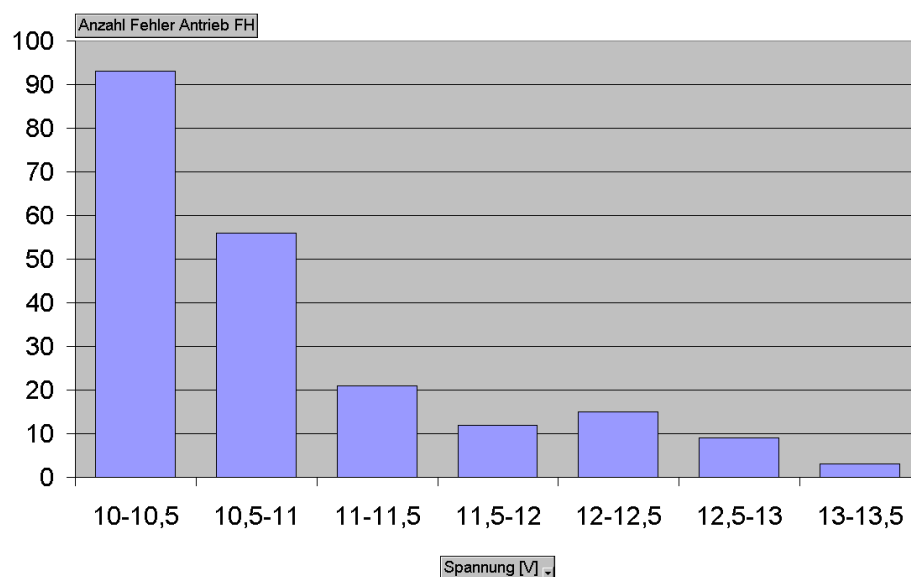


Abbildung 47: Fehler am Fensterheber im Verlauf einer Spannungsabfallsimulation

Im Rahmen dieser Untersuchung traten mit kontinuierlich abnehmender Bordnetzspannung gehäuft Fehlfunktionen an der Elektronik des Fensterhebersystems auf. Das Balkendiagramm wurde mit Hilfe einer *Pivot – Tabelle* in *Excel* auf Basis der Dokumentationsdatei der Dauererprobung erstellt. Diese Auswertung nimmt geraume Zeit in Anspruch, zudem sind die Ergebnisse nicht direkt am Prüfstandsrechner abrufbar. Dafür bedarf es einer im Prüfsystem integrierten Visualisierungsmethode zur Darstellung der Einflüsse variabler Umgebungsbedingungen.

Um einen Zusammenhang zwischen Änderungen oder Fehlfunktionen am Prüfling und den dabei vorherrschenden Umgebungsbedingungen zu untersuchen, werden durch die Prüfablaufsteuerung bei jedem abgeschlossenen Zyklus messbare Umgebungsbedingungen gespeichert. Im Weiteren bietet die Funktion der Messdatenpräsentation optional an, über dem Zyklus dargestellte Messkurven farblich in einen Bezug zu diesen Umgebungsbedingungen zu bringen. Ein Beispiel ist die Betrachtung der Schließgeschwindigkeit der Tür unter Berücksichtigung der Außentemperatur. Abbildung 48 ordnet dafür Temperaturbereichen naheliegende Farbtöne zu.

≤-20°C	-19°C bis -10°C	-9°C bis 0°C	1°C bis 14°C	15°C bis 29°C	30°C bis 44°C	45°C bis 59°C	60°C bis 74°C	≥75°C

Abbildung 48: Farbskala zur Umgebungstemperatur

Im folgenden Schritt werden die bereits in Abbildung 46 präsentierten Graphen farblich entsprechend der beim Erprobungszyklus vorliegenden Außentemperatur dargestellt.

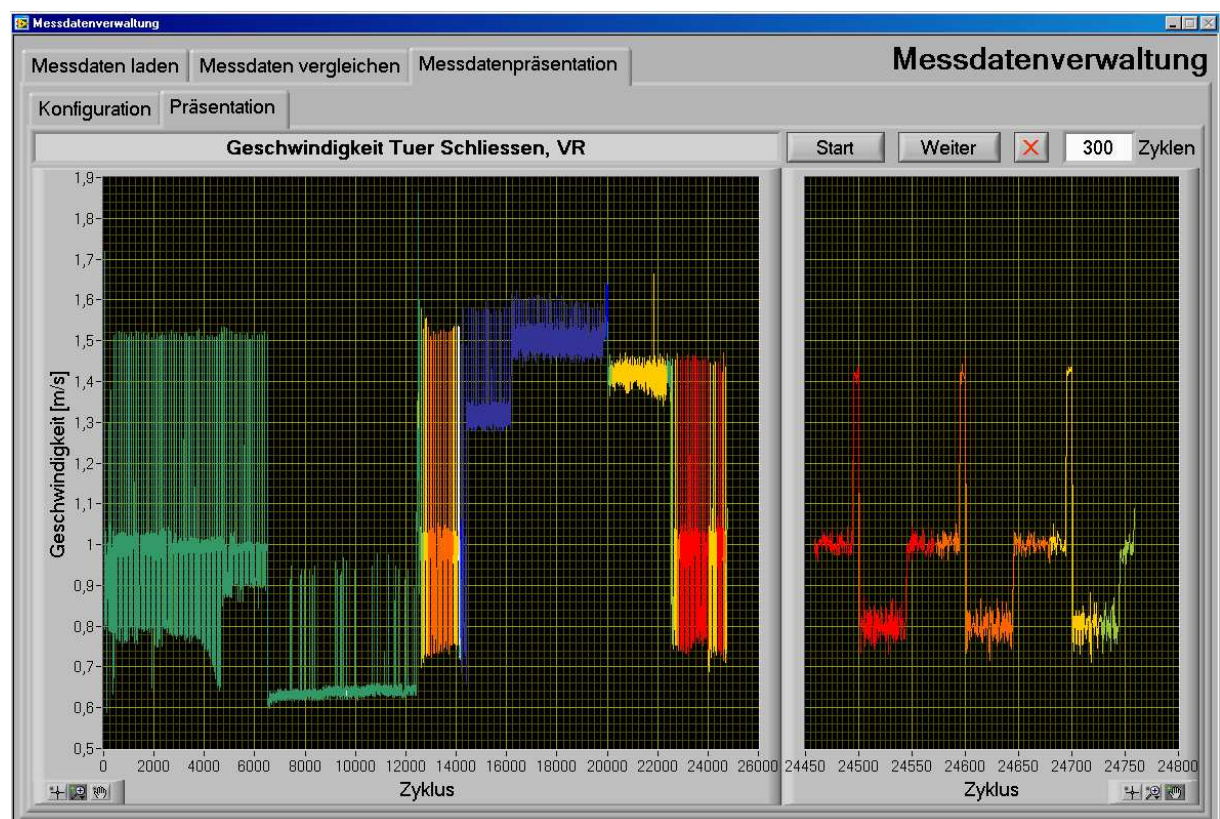


Abbildung 49: Bedienoberfläche der Messdatenpräsentation mit farblich gekennzeichneten Messdaten

Aus dem linken Graphen ist neben der Aufteilung der Temperaturzyklen während der Erprobung direkt ersichtlich, dass bei eisigen Temperaturen wesentlich erhöhte Betätigungsge-  
schwindigkeiten zum sicheren Schließen der Tür notwendig sind. Der rechte Graph veran-



schaulicht einen Temperaturabfall in der Klimakammer von mindesten 75 auf unter 45 Grad Celsius während der letzten 300 Zyklen, ohne dass dabei ein sichtbarer Einfluss auf die Türbewegung verursacht wurde.

#### 4.4.3 Überwachung zeitraffungsbedingter Effekte

Die Anforderung einer zeitgerafften Lebensdauerprobung hat aus zweierlei Gründen eine große Bedeutung. Einerseits sollen termingerecht zu den im Entwicklungszeitplan definierten Meilensteinen Ergebnisse über die bislang erzielte Qualität der Fahrzeugkomponenten vorliegen. Die Einhaltung dieser Termine wird kritisch, sobald die Erprobung durch fehlende Prüfteile oder Informationen verzögert beginnt. Zum anderen erfordert das betriebswirtschaftliche Bestreben, jedes Potential zur Einsparung von Maschinenstunden auszuschöpfen. Insbesondere Klimakammern und Shakeranlagen verursachen einen großen Anteil der Erprobungskosten. Im Gegensatz zur strengen Umsetzung dieser Anforderungen steht die Notwendigkeit, zur Sicherstellung einer realitätsnahen Erprobung, insbesondere bei elektrischen Antrieben, zwischen einzelnen Betätigungen ausreichende Pausenzeiten vorzusehen. Die Festlegung von Pausen bei Lebensdauererprobungen führt in der Praxis häufig zu Kontroversen zwischen Automobilherstellern und Zuliefererfirmen. Eine Pause von zehn Sekunden pro Zyklus führt bei 50.000 Testzyklen zu einer Verzögerung von fast sechs Tagen und verursacht dadurch hohe Zusatzkosten.

Ein optimaler Kompromiss zwischen Zeitraffung und Realitätsnähe kann im Vorfeld einer Erprobung nicht festgelegt werden. Dazu bedarf es der Überwachung und gegebenenfalls einer Nachbesserung während des Testablaufs. Am Beispiel eines Dauerversuchs zur Ermittlung von Maximalkräften beim Einklemmschutz wird dies deutlich. Abbildung 50 zeigt die momentane Krafteinwirkung auf die Fensterscheibe, bei der ein Einklemmfall erkannt wird und der Fensterheberantrieb aus Sicherheitsgründen reversiert.

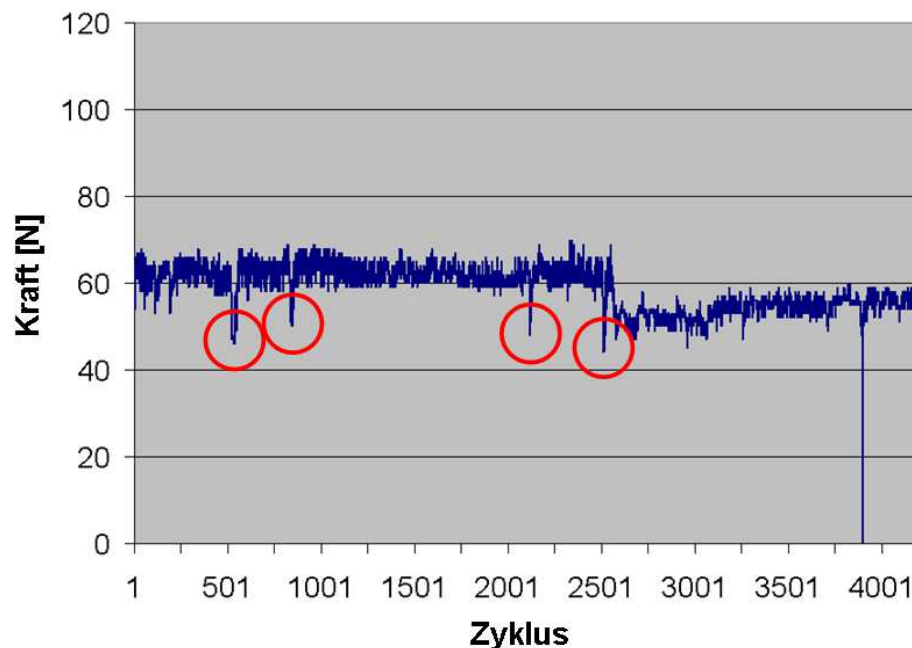


Abbildung 50: Verlauf der Einklemmkräfte am Fensterheber bei einer Dauererprobung

Die rot eingekreisten Messwerte liegen etwa 20 Prozent unterhalb der übrigen Werte bis zum Zyklus 2.500. Eine Analyse ergab, dass genau diese verminderten Kräfte aufgenommen wurden, nachdem der Dauerversuch durch prüfstandsinterne Probleme mit der Spannungsversor-



gung nach einer mehrstündigen Pause wieder gestartet worden war. Diese Pause führte durch die Abkühlung des Fensterhebermotors zu einem verfrühten Auslösen des Einklemmschutzes. Nach einer Verdoppelung der Pausenzeit zwischen den einzelnen Fensterheberbewegungen auf zwanzig Sekunden ab Zyklus 2.500 reduzierten sich die Kräfte auf ein Niveau, welches mit jenem nach den vier Ausfällen zu vergleichen ist. Dieses Beispiel zeigt den Einfluss und die Bedeutung von unsachgemäß kurzen Pausen, besonders bei sicherheitsrelevanten Komponenten.

Zur Überwachung des Kompromisses zwischen Zeitraffung und Realitätsnähe existiert im Prüfsystemkonzept das Prüfmodul *Lange Pause*. Dies lässt die Dauererprobung an definierten Zyklen für minimal 30 Minuten pausieren. Gleichzeitig wird zum jeweiligen Zyklus ein passend skaliertes  $P$  für *Pause* in sämtliche Kurven mit charakteristischen Werten eingefügt. Bei der Betrachtung dieser Messkurven in der Messdatenpräsentation ist das  $P$  sofort erkennbar.

Treten im Anschluss an eine *Lange Pause* Unregelmäßigkeiten in einer Messkurve auf, wären diese in der Darstellungsmethode aus Abbildung 51 in der Form sichtbar, wie es die Abbildung 50 zeigt. Dadurch würde ersichtlich, dass die eingestellten Pausen zwischen einzelnen Betätigungen zu kurz sind. Dies macht als Konsequenz zur Erhaltung der Realitätsnähe eine Erhöhung der Pausenzeiten erforderlich.

Im Zusammenhang mit zeitraffungsbedingten Effekten ist ebenso die Standzeit eines Prüflings unter klimatischen Bedingungen zu beachten. Ein reiner Türzuschlagstest ohne weitere Prüfmodule mit 25.000 Betätigungen in der Klimakammer kann bei einer Zykluszeit von etwa 17 Sekunden in weniger als einer Woche durchlaufen sein. Zwar werden durch die Türbetätigung die mechanischen Belastungen simuliert, für eine Reaktion des Materials auf klimatische Einflüsse sind sieben Tage jedoch keineswegs angemessen. Für eine angemessene Standzeit in der Klimakammer sind ebenso ausreichende Pausen einzuplanen.

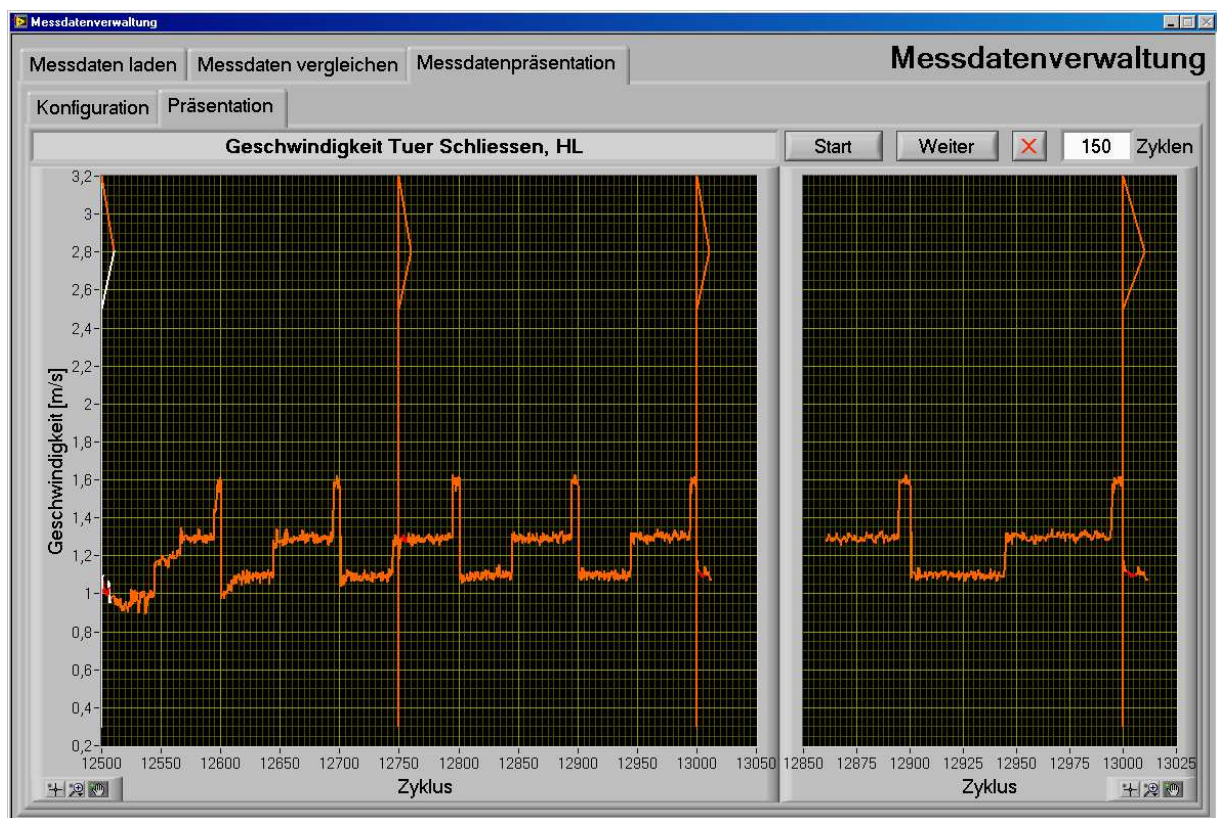


Abbildung 51: Messdatenpräsentation mit Darstellung einer *Langen Pause*

## 4.5 Verwendete Hardware

Das Kernstück des Prüfsystems ist der Prüfstandsrechner mit der daran angeschlossenen Messhardware. Die Ausführung der mechanischen Betätigung erfolgt durch Roboter. Zur Simulation von extremen Umwelteinflüssen dienen Klimakammern und Shakeranlagen.

Eine Minimierung von Steckverbindungen am Prüfstands Aufbau konnte durch die Verwendung von Robotern als Alternative zu aufwendigen Spezialgestellen, die Kommunikation mit Aktoren über Bussysteme sowie die Einbeziehung der fahrzeugeigenen Sensorik und Antriebe erreicht werden. Vormalig eingesetzte Prüfstände wiesen bei maximaler Ausstattung zirka 250 Steckverbindungen an acht Verteilerboxen und rund 100 prüfstandsinternen Endschaltern und Sensoren auf. Bei extremen klimatischen Bedingungen waren Fehler an diesen Verbindungen wiederholt der Grund für Ausfälle des Prüfsystems und langwierige Fehlersuchen. Das neue Prüfsystemkonzept besteht aus maximal 50 Steckverbindungen, von denen nur wenige unter dem Klimakammereinfluss stehen. Zusätzliche Sensoren für den Betrieb der Prüfeinrichtung werden nicht mehr benötigt.

### 4.5.1 Prüfstandsrechner

Als Prüfstandsrechner kommt ein Standard – PC mit *Intel Pentium 4* Prozessor und  $2 \times 2,4$  Gigahertz Taktrate sowie vier Gigabyte Arbeitsspeicher zum Einsatz. Zur Erfassung von 16 analogen und acht digitalen Eingängen ist die Messkarte *NI PCI-6251* installiert. Diese verfügt über einen 16 Bit Analog – Digital – Umsetzer, was einer auf den Eingangsbereich von  $\pm 10$  Volt bezogenen Auflösung von 0,3052 Millivolt entspricht. Die maximale messkarteninterne Abtastrate der einzelnen analogen Eingänge beträgt 62,5 Kilohertz. Die interne Aktualisierungsrate der digitalen Eingänge beträgt 30 Megahertz. Die geforderte Aktualisierungsrate der Datenbasis von 1.000 Hertz wird durch diese Kennwerte sichergestellt [Nat-08].



Abbildung 52: Prüfstandsrechner

Für die Beschaltung von 20 digitalen Ausgängen und zur Auswertung der vier Drehwinkelgeber ist die Messkarte *NI PCI-6601* im Einsatz. Details zur Auswertung der Drehwinkelgeber sind bereits in Abschnitt 4.3.2.2 beschrieben. Die digitale Ausgabe erfolgt nach der Ansteuerung

rung über die Prüfsystemsoftware durch einen Prozessor mit einer Taktrate von 20 Megahertz. Verzögerungen bei der Ausgabe von Schaltzuständen ergeben sich im Wesentlichen aus den Schaltzeiten angeschlossener robuster KFZ – Relais, welche typenabhängig zwischen fünf und zehn Millisekunden betragen [Nat-09c].

Für die Verbindung zur Fahrzeugelektronik dient das Hardwareinterface VN8900 der Firma Vector Informatik. Das Gerät kann wahlweise mit bis zu vier Steckmodulen der Typen Highspeed CAN, Lowspeed CAN, LIN oder FlexRay bestückt werden. Eine Eignung für die Komponentenerprobung ist in Bezug auf Flexibilität und Kanalanzahl gegeben. Die eingesetzte Rechnerhardware besteht vollständig aus Standardkomponenten, welche bei einem Ausfall binnen weniger Tage als Ersatzteil verfügbar sind. Messtechnische Bauteile sind für den Einsatz unter LabVIEW optimiert und für einen Einsatz in Prüfständen zur Dauererprobung vielfach erprobt.

Als Betriebssystem wird *Windows XP Professional Service Pack 3 (Microsoft)* verwendet. Die Installation ist minimalistisch ausgeführt. Mit Ausnahme der nicht abschaltbaren Systemdienste laufen nur die Dienste zur Überwachung der Messhardware für Netzwerkverbindungen sowie zur Verwaltung logischer Datenträger. Letztgenannte sind notwendig, um Daten über USB – Speichermedien auszutauschen. Die Installation von Sicherheitssoftware, beispielsweise ein Virenschutzprogramm, ist aus Gründen der Performance nicht möglich. Der Rechner ist daher netzwerktechnisch nicht in die Firmeninfrastruktur eingebunden. Die beschriebene Ausstattung und Konfiguration ermöglicht es, die Prüfsystemsoftware im Rahmen der *weichen* Echtzeitanforderungen zu betreiben.

Zu Demonstrationszwecken oder für Prüfungen ohne relevante Messdatenerfassung ist die Dauerlaufsoftware auf jedem Rechnersystem mit LabVIEW – Laufzeitumgebung ausführbar. Die Kanäle zur Messdatenerfassung können über die Bedienoberflächen der jeweiligen Komponenten angepasst werden. Die *weiche* Echtzeitfähigkeit ist nicht mehr gegeben, eine Anhäufung von Timeout – Fehlern in der Ereignisanzeige ist deutlich erkennbar. Diese Konzeption führt zu einer sehr flexiblen und ortsunabhängigen Verwendbarkeit des Prüfsystems. Der durchgehende Einsatz von Standardkomponenten sorgt für eine hohe Zuverlässigkeit und erfüllt den Wunsch nach überschaubaren Investitionskosten für die Computertechnik.

## 4.5.2 Robotik

### 4.5.2.1 Einsatz in der Komponentenerprobung

Versuche mit mechanischen Betätigungen benötigen ohne den Einsatz der Robotik einen großen Zeitaufwand für Aufbau und Inbetriebnahme. Zur Positionierung von Aktoren und zur Adaption der Weg- oder Winkelaufnehmer müssen Gestelle und Halterungen aufgebaut werden. Für die Ausführung von Kurvenbahnen sind aufwendige Konstruktionen erforderlich. Bei einem Einsatz pneumatischer Antriebe unter extremen Klimabedingungen ist die Einstellung und Einhaltung konstanter Geschwindigkeiten sehr mühsam. In der Praxis hat sich gezeigt, dass einmal aufgebaute Spezialvorrichtungen nach Beendigung eines Versuchs nur äußerst selten ein zweites Mal in gleicher Weise Verwendung finden. Da ein Versuch die hohen Anschaffungskosten nicht abdeckt, werden Einzelteile oftmals für andere Tests entnommen. Beispiele dafür sind bereits in Abbildung 11 sowie in Abbildung 53 dargestellt.

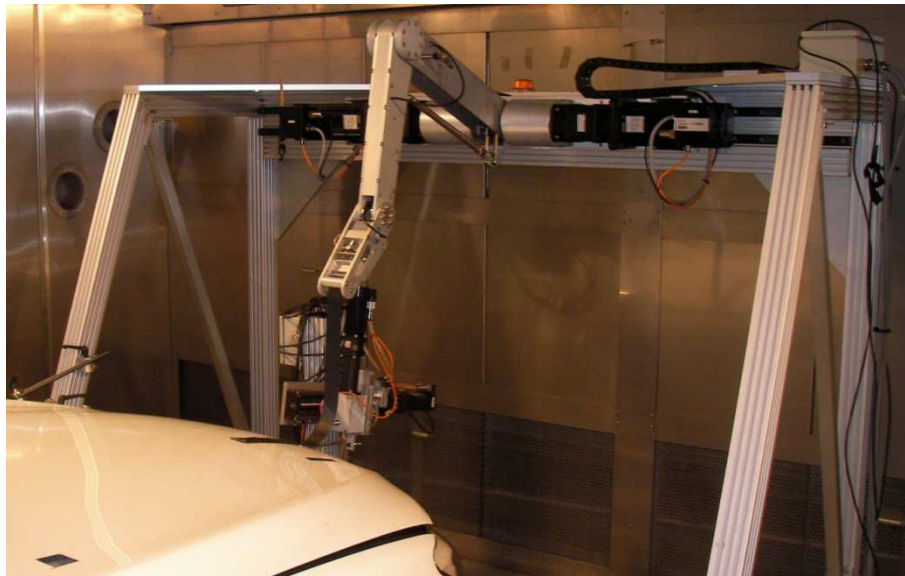


Abbildung 53: Spezialvorrichtung zur Klappenbetätigung

Zur Optimierung der Versuchsaufbauten sind Standard – Industrieroboter aufgrund ihrer mechanischen Freiheitsgrade für die Komponentenerprobung prädestiniert. Der Aufbau von Spezialgestellen zur Ausrichtung der Aktoren kann dadurch in der Regel entfallen, Kreisbewegungen werden problemlos ermöglicht. Die Geschwindigkeiten und Beschleunigungen sind in feinen Stufen einstellbar, eine integrierte Wegmessung ist realisierbar. Kurzfristige Änderungen am Prüfaufbau als Reaktion auf Bauteilveränderungen lassen sich bei Robotern optimal umsetzen. Weitere Beispielanwendungen finden sich in den Quellen [Bat-05], [Bat-08] und [Schü-10]. Aufgrund der Verwendung von Industrierobotern können eine Reihe von Anforderungen aus Tabelle 2 eindeutig erfüllt werden. Dazu zählen die Art der Energieeinleitung für realitätsnahe Belastungen, die Effizienz bei der Inbetriebnahme von Prüfungen sowie die Einstellung flexibler Bewegungsabläufe. Im Übrigen stehen für Industrieroboter als Standardkomponenten Ersatzteile binnen weniger Tage zur Verfügung. Eine Reparatur kann direkt vor Ort erfolgen.

#### 4.5.2.2 Vergleich von Robotermodellen

Die Auswahl eines optimalen Robotersystems für Komponentenversuche basiert auf zwei übergeordneten Kriterien. Als erstes muss der Roboterarm so beschaffen sein, dass er ausreichende Reichweite, Traglast, Geschwindigkeit und Bewegungsfreiheit bietet. Die Werte können den Datenblättern der einzelnen Robotertypen entnommen werden. Zweitens sind die Rechengeschwindigkeit, die Schnittstellen und der Funktionsumfang zu betrachten. Diese Eigenschaften entscheiden darüber, ob sich der Roboter als Aktor mit integrierter Messdatenverarbeitung für den Einsatz in der Komponentenerprobung eignet. Konkrete Informationen zur Verarbeitungsgeschwindigkeit von Befehlen auf dem Robotercontroller sind weder aus Datenblättern noch durch anderweitige Herstellerinformationen zuverlässig zu beschaffen. Eine Bewertung der Fähigkeiten einzelner Robotersysteme ist nur durch umfangreiche Untersuchungen möglich.





Zu Beginn der Entwicklung dieses Prüfsystemkonzepts standen bereits Roboter vom Typ *Mitsubishi RV-12SL* zur Verfügung. Dieses Modell verfügt bei einer Traglast von 12 Kilogramm über ein vergleichsweise geringes Eigengewicht von 98 Kilogramm, was die Handhabung bei der Inbetriebnahme von Versuchsaufbauten stark erleichtert. Allerdings zeigte sich, dass die Reichweite von 1.385 Millimetern bei Tür- und Klappenbetätigungen grenzwertig ist. Bei der-

artigen Betätigungen muss der Roboter außerhalb des Bewegungsbereichs der Tür oder Klappe positioniert sein, trotzdem aber die Kraftangriffspunkte erreichen. Es zeigte sich ebenfalls, dass der maximale Bewegungswinkel der vierten Achse bei einer Nachstellung menschlicher Bewegungen der entscheidende Faktor ist. Dieser weist beim *RV-12SL* relativ geringe  $\pm 160$  Grad auf.





Neben den mechanischen Einschränkungen hat sich auch der Robotercontroller als ungeeignet herausgestellt. So bietet die Programmiersprache *MELFA Basic IV* keine Funktion an, welche die aktuelle Position des Werkzeugs zur Wegerfassung während des Bewegungsablaufs kontinuierlich exportiert. Die einzige Möglichkeit zur Erzeugung von Wegdiagrammen besteht darin, die Berechnung der Bahnkurve auf einen externen Rechner zu verlagern und von dort die Positionsdaten in Echtzeit an den Robotercontroller zu übertragen. Eine derartige Ansteuerungsmethode würde jedoch einen zu großen Aufwand bedeuten und einem praktischen Einsatz in der Komponentenerprobung entgegenstehen.

Aus den genannten Gründen musste im Rahmen dieses Prüfsystemkonzepts ein alternatives Robotersystem ausgewählt werden. Dabei stehen besonders eine Erhöhung der Reichweite, die Bewegungsfreiheit der vierten Achse, möglichst hohe Winkelgeschwindigkeiten der ersten drei Achsen sowie ein Roboterarm mit minimalen Störkonturen im Vordergrund. Der letztgenannte Aspekt begünstigt besonders die Installation einer Roboterschutzhülle. Tabelle 5 zeigt eine Übersicht der Modelle, die aus mechanischer Sicht als Alternativen möglich sind.



				
H	ABB	Adept	COMAU	FANUC
M	IRB 2600 [Abb-11]	Viper s1700 [Ade-10]	NS 16 [Com-07]	M-20iA [Fan-10]
R	1650mm	1717mm	1650mm	1811mm
L/E	20kg / 280kg	20kg / 280kg	16kg / 335kg	20kg / 250kg
W4	$\pm 400^\circ$	$\pm 200^\circ$	$\pm 270^\circ$	$\pm 200^\circ$
G1-3	175°/s	170°/s, 170°/s, 175°/s	155°/s, 155°/s, 170°/s	195°/s, 175°/s, 180°/s

				
H	Kawasaki	KUKA	Mitsubishi	Motoman
M	FS20N [Kaw-08]	KR 16-2 S [Kuk-09]	RV-12SDL [Mit-10]	HP20 [Mot-08]
R	1650mm	1611mm	1385mm	1717mm
L/E	20kg / 280kg	16kg / 235kg	12kg / 98kg	20kg / 280kg
W4	$\pm 200^\circ$	$\pm 350^\circ$	$\pm 160^\circ$	$\pm 200^\circ$
G1-3	160°/s, 140°/s, 160°/s	168°/s, 173°/s, 192°/s	230°/s, 172°/s, 200°/s	170°/s, 170°/s, 175°/s



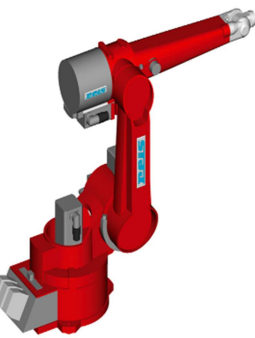

				
H	Nachi	OTC	Reis	Stäubli
M	MC20-01 [Nac-08]	A2-V20 [Otc-09]	RV20-16 [Rei-06]	RX160L [Stä-11a]
R	1722mm	1710mm	1500mm	2010mm
L/E	20kg / 230kg	20kg / 285kg	16kg / 220kg	28kg / 250kg
W4	$\pm 180^\circ$	$\pm 180^\circ$	$\pm 210^\circ$	$\pm 270^\circ$
G1-3	170°/s	195°/s, 190°/s, 180°/s	200°/s, 165°/s, 150°/s	200°/s, 200°/s, 255°/s

Tabelle 5: Vergleich von Roboterarmen zur Türbetätigung

Legende: H: Hersteller, M: Modell, R: Reichweite, L: Traglast, E: Eigengewicht, W4: Maximaler Drehwinkel der Achse 4, G1-3: Maximale Winkelgeschwindigkeiten der Achsen 1 – 3.

Tabelle 6 liefert eine vergleichende Bewertung der Kriterien.

	Reichweite	Traglast	Eigengewicht	Drehwinkel Achse 4	Winkelgeschwindigkeit Achse 1 - 3	Beschaffenheit für Schutzhülleneinsatz
ABB IRB 2600	o	o	-	+	o	o
Adept Viper s1700	o	o	-	o	-	o
COMAU NS 16	o	o	o	+	-	o
FANUC M-20iA	+	o	o	o	o	o
Kawasaki FS20N	o	o	-	o	-	+
KUKA KR 16-2 S	o	o	o	+	-	o
Mitsubishi RV-12SDL	-	-	+	-	-	+
Motoman HP20	o	o	-	o	-	o
Nachi MC20-01	o	o	o	-	-	o
OTC A2 – V20	o	o	-	-	o	o
Reis RV20-16	o	o	o	o	-	o
Stäubli RX 160L	+	+	o	+	+	+

**Tabelle 6: Zusammenfassender Vergleich der Roboterarme**

Aus der Betrachtung der Tabellen 5 und 6 wird deutlich, dass der Roboterarm des *Stäubli RX160L* bestmöglichst für den Einsatz in der Komponentenerprobung geeignet ist. Die Forderungen bezüglich der Reichweite und Traglast werden erfüllt, das Eigengewicht ist unter Berücksichtigung der verfügbaren Flurförderzeuge in Werkstätten zulässig. Die Winkelgeschwindigkeiten sind im Vergleich die höchsten und reichen erfahrungsgemäß zur Realisierung der geforderten Betätigungsgeschwindigkeiten aus. Der Roboterarm verfügt in dieser Kategorie über die größten Kennwerte, daher sind die gewünschten Lastreserven vorhanden.

Eine analoge Bewertung der jeweiligen Robotercontroller ist aufgrund von fehlenden zuverlässigen Angaben in den Datenblättern nicht möglich. Die Veröffentlichung [Her-10], mehrere Fachgespräche mit unabhängigen Wissenschaftlern der Firma *RTLeaders* sowie Diskussionen im Roboterforum [www.roboterforum.de](http://www.roboterforum.de) kommen zu dem Ergebnis, dass der Robotercontroller *CS8C* der Firma *Stäubli* aktuell die leistungsfähigste Steuerung am Markt ist. Daher erscheint dieser, besonders im Zusammenhang mit der Bewertung der Roboterarme, am besten für die Anforderungen des Prüfsystemkonzepts geeignet. Aus den Schulungsunterlagen der Firma *Stäubli* geht hervor, dass der multitaskingfähige Rechner des Robotercontrollers *CS8C* in einem Arbeitstakt von einer Millisekunde bis zu 3.000 Befehle der *C* – ähnlichen Programmiersprache *VAL3* ausführen kann [Stä-09]. Befehle und Funktionen zum Auslesen der Werkzeugposition sowie zur Weiterleitung an den Prüfstandsrechner sind vorhanden. Tests im Rahmen dieser Entwicklung haben ergeben, dass mit der *CS8C* – Steuerung ein Messdatenexport mit 500 Hertz über die Ethernetschnittstelle möglich ist. Für die Umsetzung dieses Prüfsystemkonzepts werden daher Roboter der Firma *Stäubli* verwendet. Weitere Details zum Programmcode auf dem Robotercontroller folgen im Abschnitt 4.6.4.

#### 4.5.2.3 Arbeitssicherheit

Wie bereits in Abschnitt 4.1.2.8 erläutert, hat die Arbeitssicherheit im Zusammenhang mit der Verwendung von Industrierobotern oberste Priorität. Softwareseitig sind daher keine Vorkehrungen in das Prüfsystemkonzept integriert, da der Not – Aus – Kreislauf unabhängig und übergeordnet mit harten Echtzeitanforderungen umzusetzen ist. Bei einem Einsatz des Prüfsystems in Klimakammern erfolgt über die Kontakte der Zugangstüren eine automatische Abschaltung der Roboterbewegung. Für Versuche außerhalb der Klimakammer eignen sich ortsflexible Mehrstrahl – Sicherheits – Lichtschranken, die beispielsweise von der Firma *Sick* angeboten werden [Sic-10].

Ein alternativer Ansatz für einen flexiblen Zugang zum Gefahrenbereich sieht vor, je nach Abstand zwischen Mensch und Roboter die Geschwindigkeit bis zum Stillstand zu drosseln [Mau-07]. Weitere Varianten auf Basis einer Kameraüberwachung sind in den Quellen [Beu-05], [Thi-05], [Hel-06] und [Mat-06] beschrieben. Diese Alternativen sind im Gegensatz zu den Lichtschranken nicht ausreichend flexibel und daher untauglich für den Einsatz im Werkstattalltag.

#### 4.5.2.4 Einsatz unter extremen Klimabedingungen

Die Prüfvorschriften aller Automobilhersteller schreiben für die Erprobung von beweglichen Karosseriekomponenten anteilmäßige Betätigungszyklen unter extremen klimatischen Bedingungen vor. Der Temperaturbereich erstreckt sich hierbei von -40 bis +80 Grad Celsius. Die relative Luftfeuchte beträgt bis zu 95 Prozent. Für Roboterarme stellen diese Umweltbedingungen eine große Belastung dar. Derzeit sind am Markt keine Roboter verfügbar, die ohne spezielle Schutzhüllen unter diesen extremen Belastungen einsetzbar sind.

Zur Erhaltung der Bewegungsfreiheit und damit der Flexibilität von Robotern im praktischen Einsatz sind bei der Schutzhüllenentwicklung gewisse Auflagen einzuhalten. Die Schutzhülle muss im Bereich der Roboterachsen 1, 4 und 6 mit Lagern ausgestattet sein, die eine vollständige Drehbarkeit dieser Achsen gewährleisten. Für die Achsen 2 und 3 genügt ein maximaler Drehwinkel von  $\pm 90$  Grad, ausgehend von ihrer Grundstellung. Im vorderen Bereich des Roboterarms sind die Platzverhältnisse durch die Anordnung der Achsen 5 und 6 sehr beengt. Eine maximale Drehbarkeit der Achse 5 ist nur bei ausreichendem Volumen in der Schutzhülle gegeben. Zur Erhaltung der ortsflexiblen Einsatzmöglichkeit muss an der Hülle eine Durchführung für die Versorgungskabel vorhanden sein. Durch Bewegungen des Roboters ist es unvermeidbar, dass sich Teile der Außenhaut gegeneinander reiben. Mit einer entsprechend gewählten Beschichtung lassen sich reibungsbedingte Beschädigungen verhindern. Ebenso garantiert ein Überdruck in der Schutzhülle, dass keine Feuchtigkeit von außen an den Roboterarm gelangt. Die Durchflutung der Schutzhülle mit Luft zum Erhalt der roboterseitig zulässigen Temperaturen wird mit Hilfe eines Mitteldruckgebläses sichergestellt, welches genügend Luftstrom erzeugen kann. Die detaillierte Entwicklung und Produktion ist nicht Bestandteil dieser Arbeit, damit wurden Spezialisten beauftragt. Durch den Einsatz der Schutzhüllen ist die Funktionalität der Roboter selbst unter extremen Umweltbedingungen gewährleistet.





Abbildung 54: Robotersystem *Stäubli RX160L* mit Klimaschutzhülle

### 4.5.3 Spannungsversorgung

Für eine zuverlässige Erprobung elektronischer Systeme ist die Sicherstellung einer intakten Energieversorgung von elementarer Bedeutung, da die Steuergeräte nur dann einwandfrei arbeiten können [Rei-08]. In der Vergangenheit führten eine improvisierte Verkabelung und zu gering dimensionierte Netzteile zu sporadischen und selten reproduzierbaren Fehlfunktionen, beispielsweise Reversieren am Fensterheber. Für den Aufbau eines Versuchs ist daher die Verwendung von Original – Fahrzeugkabelbäumen ein wesentliches Kriterium. Erweiterungen dieser Kabelsätze sind notwendig, sobald prüfungsrelevante Leitungen in der vorliegenden Ausbaustufe nicht berücksichtigt wurden. Ergänzungen sind dann mit den vorgeschriebenen Kabellängen, Querschnitten und Steckverbindungen vorzunehmen. Als Spannungsquelle dient eine neuwertige Autobatterie, welche am dafür vorgesehenen Einbauort im Fahrzeug zu platzieren ist. Die Batterie sorgt für eine ausreichende Reserve bei der Energieversorgung. Die Batteriespannung wird mit Hilfe eines Batterie – Ladecomputers durch Laden oder Ladungserhaltung auf dem in der Prüfvorschrift definierten Niveau sichergestellt.

Im Prüfsystem kommt der Batterie – Ladecomputer *DBL800* der Firma *Deutronic* zum Einsatz. Dieses Gerät verfügt über diverse Sicherheitsfunktionen, zum Beispiel Kurzschlusserkennung, Verpolschutz oder Ladestrombegrenzung. Ladecomputer der Firma *Deutronic* werden bei allen namhaften Automobilherstellern in der Produktion eingesetzt, um eine optimale Spannungsversorgung der Fahrzeugelektronik beim *Flashen* der Steuergeräte zu gewährleisten [Deu-10].

#### 4.5.4 Umweltsimulation

Lebensdauererprobungen von Karosserieteilen erfolgen anteilig in Klimakammern und auf Shakeranlagen. Die üblicherweise sehr extrem simulierten Umweltbedingungen dienen als Kompromiss für die Durchführung der Tests mit hoher Zeitraffung. Gerätschaften zur Umweltsimulation sind in der Regel eigenständige Prüfstände, sie werden für Komponentenversuche mit dem in dieser Ausarbeitung entwickelten Prüfsystemkonzept verknüpft. Dies ermöglicht die Aufnahme der Umweltbedingungen als Messdaten sowie eine vom Prüfablauf abhängige Parametrierung der Anlagen. Bei Störungen oder Erreichen eines definierten Zielzyklus können die sehr energieintensiven Kammern oder Shaker zur Kosteneinsparung automatisch abgeschaltet werden.

Für die Erprobung von beweglichen Karosserieteilen ist darauf zu achten, dass in den Kammern genügend Platz zur Verfügung steht. Ein Türzuschlagsversuch setzt mindestens einen Meter Freiraum an beiden Fahrzeugseiten voraus. Die Verwendung der in Abschnitt 5.1.3 vorgestellten Hilfsmittel zur Adaption der Roboter am Fahrzeug erfordert eine ausreichende Traglast der Böden. Für die Versorgungsleitungen der Roboterarme und Schutzhüllen sind pro Robotersystem mindestens zwei Durchführungen notwendig. Um den Datenaustausch mit der Prüfsystemsoftware gewährleisten zu können, müssen die Steuereinheiten der Geräte zur Umweltsimulation über geeignete Schnittstellen, beispielsweise OPC, verfügen.



Abbildung 55: Versuchsaufbau einer Türerprobung in der Klimakammer

#### 4.5.5 Messmittel

Die folgenden Abschnitte führen exemplarisch Messmittel auf, die sich bei Dauererprobungen im praktischen Einsatz bewährt haben. Jeder Sensor erfüllt mit seinen Spezifikationen die geforderten Temperatur- und Feuchtigkeitsbedingungen und ist als Standardkomponente ohne Sonderausführung am Markt erhältlich. Trotzdem resultiert aus den extremen Umweltbedingungen für die Messmittel eine erhöhte Gefahr von Fehlfunktionen oder gar Ausfällen. Daher sind im Prüfsystem Funktionen zur Überwachung und für Plausibilitätsprüfungen implementiert.

#### 4.5.5.1 Drehwinkelgeber

Drehwinkelgeber werden eingesetzt, um Winkel und Winkelgeschwindigkeiten von drehbar gelagerten Bauteilen, wie Türen oder Klappen, zu messen. Bewährt hat sich das Modell *Sendix inkremental Typ 5020* der Firma *Kübler*. Die Auflösung ist mit 3.600 Impulsen pro Umdrehung ausreichend genau und der Temperaturbereich erstreckt sich von -40 bis +85 Grad Celsius. Die Schutzart *IP67* ermöglicht einen Einsatz bei hoher Luftfeuchtigkeit [Küb-10]. Die Gefahr einer Wasserdiffusion in den Innenraum und damit in die sensible Elektronik des Geräts, was in einem konkreten Fall zum Defekt des Drehgebers führte, wird durch Anschlusskabel ohne Steckverbindungen minimiert. Zur Adaption der Drehwinkelgeber am bewegten Bauteil kommt eine Mechanik aus zwei miteinander verbundenen Parallelogrammen zum Einsatz, welche die rotatorische und translatorische Bewegung des Anschraubpunkts am Bauteil entkoppelt. Dies ist notwendig, da der Drehwinkelgeber in der Regel nicht direkt an der Drehachse des rotierenden Bauteils adaptiert werden kann. Für eine permanente Überwachung der Funktionalität und Plausibilität des Sensors ist ein Abgleich des Winkelmaßes an definierten Winkelstellungen des Prüflings oder ein Vergleich der Winkelgeschwindigkeit mit Grenzwerten zweckmäßig. Fehler in der Zählweise oder komplette Funktionsausfälle lassen sich dadurch zuverlässig erkennen.

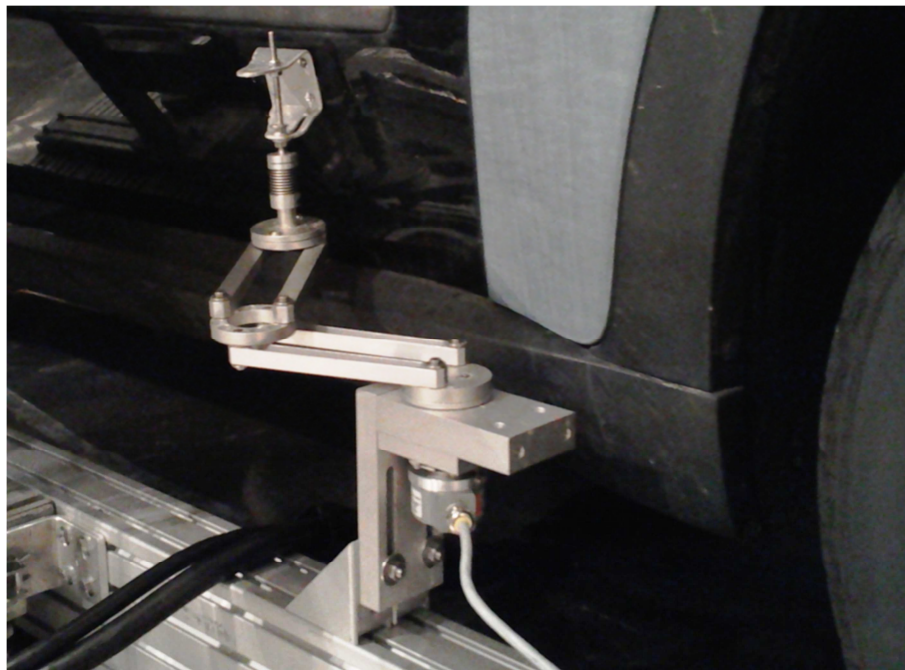


Abbildung 56: Drehwinkelgeber an der Beifahrertür

#### 4.5.5.2 Näherungssensor

Näherungssensoren dienen dazu, berührungslos Positionsabfragen von beweglichen Karosserieelementen auszuführen. Im praktischen Einsatz eigneten sich induktive Sensoren der Serie *E2A* der Firma *OMRON* sehr gut. Sensoren dieser Serie bieten die Schutzart *IP67*, sind für einen Temperaturbereich von -40 bis +85 Grad Celsius ausgelegt und arbeiten mit einer internen Aktualisierungsrate von 1.000 Hertz. Eine direkt am Anschlussstecker platzierte LED unterstützt die Justierung des Abstands zum Prüfling durch ein stufenloses Außengewinde. Zu beachten ist, dass der Schaltabstand unter Temperatureinfluss um  $\pm 15$  Prozent variieren kann. [Omr-08].



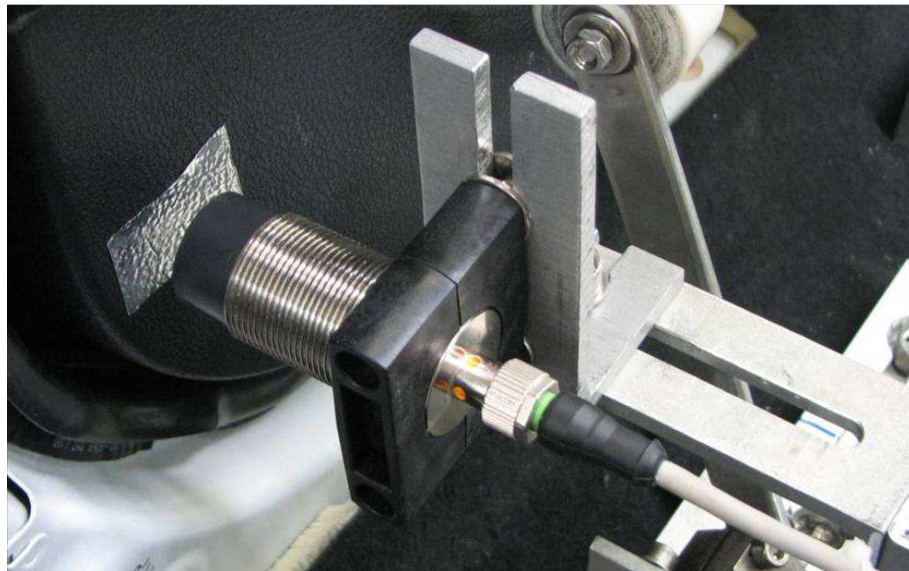


Abbildung 57: Näherungssensor an der Türinnenverkleidung

#### 4.5.5.3 Stromsensor

Mit Hilfe von Stromsensoren erfolgen Messungen von Stromverläufen sowie die Detektierung von Endlagen bei elektrischen Antrieben durch Blockstromerkennung. Im praktischen Einsatz hat sich der auf dem Halleffekt basierende Stromsensor *HASS 50-S* der Firma *LEM* bewährt. Der Sensor kann aufgrund seiner sehr geringen Ausmaße einfach an Messstellen im Fahrzeug, häufig in direkter Umgebung der Steuergeräte oder Antriebe, platziert werden. Der zulässige Temperaturbereich erstreckt sich von  $-40$  bis  $+85$  Grad Celsius, der Messbereich beträgt  $\pm 50$  Ampere bei einer Genauigkeit von  $\pm 1$  Prozent [Lem-11].

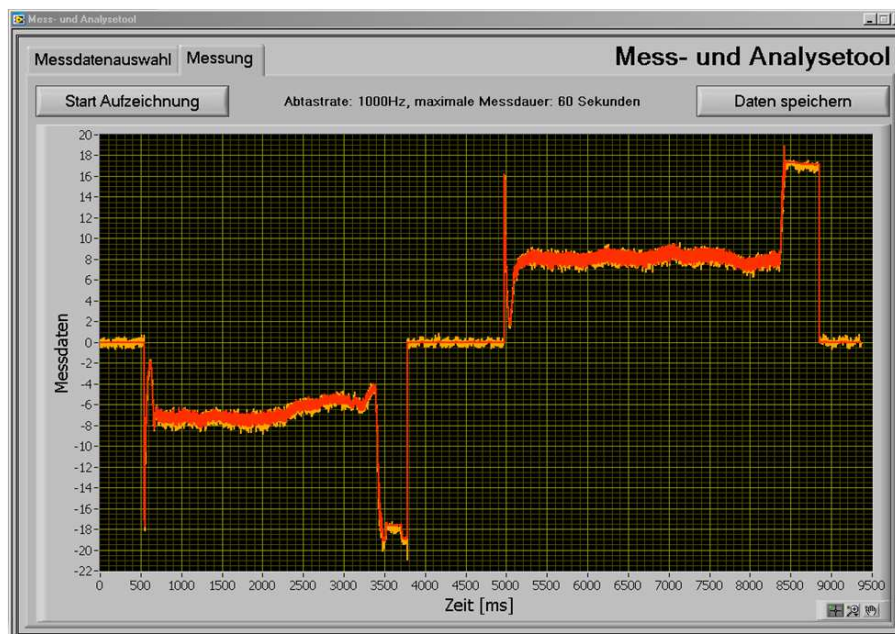


Abbildung 58: Vergleichsmessung zwischen Strommesszange *HZ56-2* und Stromsensor *HASS 50-S*

Bezogen auf übliche Stromstärken von drei bis zehn Ampere während der Betätigung elektrischer Antriebe von Karosseriekomponenten reicht diese Genauigkeit für Vergleichsmessungen nicht aus. Dafür sind weniger streuende Messmittel zu verwenden, wie die Strommesszange *HZ56-2* des Herstellers *HAMEG*, die aber nicht für den Einsatz in Klimakammern ge-

eignet ist. Für automatisierte Strommessungen zur Energieberechnung oder Blockstromerkennung reicht die Genauigkeit des robusten Sensors *HASS 50-S* aus. Abbildung 58 stellt beide Stromsensoren in einer Vergleichsmessung gegenüber. Bei der orange dargestellten Messkurve des *HASS 50-S* ist die erhöhte Streuung besonders im Leerlauf und bei Blockströmen erkennbar. Die rot angezeigten Messdaten der Strommesszange *HZ56-2* sind in den Bereichen äußerst genau.

#### 4.5.5.4 Kraftsensor

Kraftsensoren dienen im Prüfsystem der Messung von Betätigungskräften an beweglichen Bauteilen sowie zur Ermittlung des Verformungswiderstands bei Steifigkeitsuntersuchungen. Piezoelektrische Sensoren der Firma *Kistler* zeichnen sich durch eine kompakte Bauweise, einen Temperaturbereich von -40 bis +120 Grad Celsius sowie durch die Schutzart *IP67* aus. Einachsige piezoelektrische Kraftsensoren reagieren jedoch sehr empfindlich auf Biegemomente und Schubkräfte, daher ist die Kraftmessdose nach dem Prinzip einer Pendelstütze an beiden Enden gelenkig zu lagern. Die vom Kraftmesselement abgegebene Ladung steigt proportional zur wirkenden Kraft und muss zur Weiterverarbeitung im Prüfsystem in eine Spannung umgewandelt werden. Für diesen Zweck bietet *Kistler* die Ladungsverstärker *5073A* sowie *5015A* an, diese sind über eine serielle RS – 232 Schnittstelle programmierbar. Entscheidend ist bei den Geräten die Funktion zur schnellen Datenübertragung. Dadurch ist es möglich, kontinuierlich Messdaten in digitaler Form mit bis zu 1.000 Hertz auszugeben und zur Kraft – Weg – Synchronisation an einen Robotercontroller zu übermitteln [Kis-10a]. Die klimatauglichen Sensorkabel zwischen Messelement und Ladungsverstärker haben sich im praktischen Einsatz als Schwachstelle herausgestellt. Bereits kleinste Beschädigungen durch Knickung führen zu starken Drifts in den Messwerten des Kraftverlaufs. Da die Messelemente beim Beginn und am Ende einer Messung unbelastet sind, eignet sich zur Kontrolle ein Vergleich des ersten und letzten Messwerts. Entsteht innerhalb einer Messdauer von 20 Sekunden eine Abweichung von mehr als drei Newton, liegt mit hoher Wahrscheinlichkeit ein Fehler in der Messkette vor.

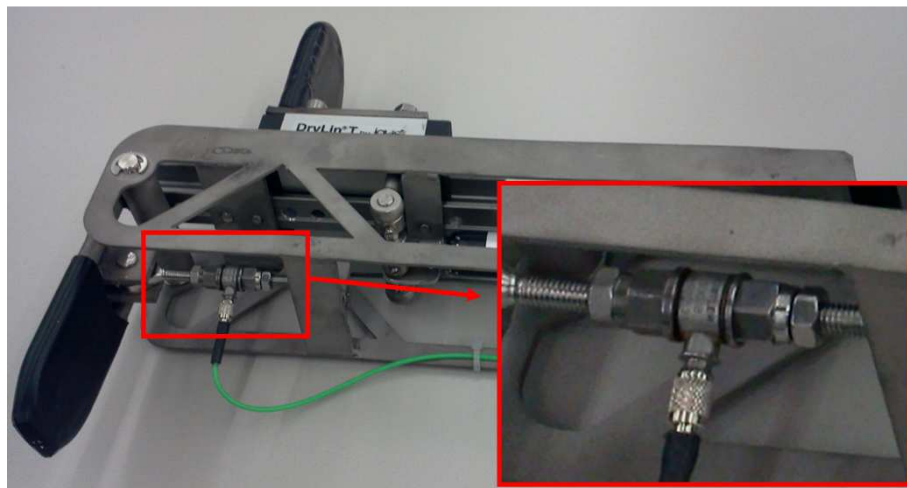


Abbildung 59: Kraftmesselement im Roboterwerkzeug

#### 4.5.5.5 Roboterbasierte Wegmessung

Bei der Ausführung einer Bewegung durch den Roboterarm dient dessen integriertes Messsystem zur Aufnahme von Wegen. Dazu erfolgt die kontinuierliche Übertragung der Positionsdaten des *Tool Center Points* im Roboterkoordinatensystem mit 500 Hertz an den Prüf-

standsrechner. Über die Änderungen der Koordinaten lässt sich der zurückgelegte Weg berechnen. Weitere Details folgen in Abschnitt 4.6.3.1.

## 4.6 Aufgabenorientierte Schnittstelle zur Robotik

Der entscheidende Faktor für den wirtschaftlich rentablen Einsatz von Industrierobotern im Versuchswesen sowie allgemein in kleinen und mittelständischen Unternehmen ist eine den spezifischen Erfordernissen angepasste Programmierungsumgebung. Die seitens der Roboterhersteller mitgelieferten Programmierwerkzeuge sind für Versuchsingenieure ungeeignet, da es ohne entsprechende Vorkenntnisse und Erfahrungen kaum möglich ist, typische Bewegungsabläufe der Komponentenerprobung mit integrierter Messdatenerfassung einzustellen. Eine Qualifizierung der Mitarbeiter durch Schulungen ist nur sinnvoll, wenn im Anschluss eine permanente Anwendung der Kenntnisse in der Praxis erfolgt. Durch wechselnde Projekte und eine schwankende Auslastung im Versuchswesen sind diese Voraussetzungen nicht gegeben.

Der Lösungsansatz für diese Problematik besteht darin, zwischen dem Robotersystem und dem Bediener eine zusätzliche Abstraktionsschicht einzufügen. Diese Abstraktionsschicht stellt Befehle und eine Bedienoberfläche zur Verfügung, durch die der Roboter in einem abgegrenzten Umfeld unter Einbeziehung der dort typischen Sensorik ohne Kenntnisse und Erfahrungen in der herstellerspezifischen Programmiersprache bedient werden kann. Maßgeblich für den Erfolg dieser Schnittstelle ist die Eigenschaft, dass mit verhältnismäßig wenigen Funktionen und Parametern eine Flexibilität erreicht wird, die eine Umsetzung aller anfallenden Aufgaben ermöglicht. Ein universelles Programmiersystem für alle Anwendungen im Bereich der Robotik ist weder wirtschaftlich noch technisch die optimale Lösung. Jedes Anwendungsgebiet erfordert speziell zugeschnittene Lösungen. Vergleichbare theoretische Ansätze werden in den Quellen [Dam-96], [Vog-99], [Bre-05] und [Her-08] beschrieben. Die in dieser Arbeit entwickelte aufgabenorientierte Schnittstelle ist auf die Einbindung von Robotik in Prüfsysteme zugeschnitten.

### 4.6.1 Anforderungen

Zur Einstellung eines Bewegungsablaufs ist eine übersichtliche und intuitiv bedienbare Oberfläche nötig. Einstellungen sowie Änderungen über das herstellerspezifische Programmierwerkzeug sind nicht vorgesehen. Lediglich die manuelle Bewegung des Roboterarms und Eingaben über das Handbediengerät lassen sich mit den geringen Qualifikationsanforderungen an die Bediener vereinbaren. *Einfachste Betätigungen*, beispielsweise das kontinuierliche Drücken eines Schalters am Lenkrad, sollen ohne die Unterstützung von Roboter- oder Softwareexperten durch Versuchsingenieure einstellbar sein.

Mit der aufgabenorientierten Schnittstelle müssen Bewegungsabläufe zu realisieren sein, bei denen das Teach – In – Verfahren üblicherweise an Grenzen stößt. Dies betrifft insbesondere kraftgeregelte Betätigungen oder zerstörende Prüfungen, da hierbei eine variable und nicht fest definierbare Endposition der Bewegungen vorliegt. Bei Druckversuchen befindet sich die Endposition inmitten des Bauteils und ist daher nicht zugänglich.

Die Abnutzung durch dauerhafte Betätigungen sowie extreme Umweltbedingungen verursacht ständige Veränderungen an Prüflingen. Ein Beispiel ist der benötigte Weg zum Entriegeln einer Tür am Türaußengriff. Dieser Weg verlängert sich durch die Ausdehnung des Bow-

denzugs sowie durch Nachgiebigkeiten und Abnutzungen beim Material. Notwendige Anpassungen der Roboterbewegung zur Vermeidung von Stillständen im Prüfablauf müssen über Parameter möglich sein, ohne die Klimakammer zum Nachteachen betreten zu müssen. Ebenso sind automatisierte Anpassungen dieser Parameter auf Basis von Messdaten zweckmäßig.

Mit Hilfe der Schnittstelle ist die Integration einer bereichstypischen Sensorik ins Robotersystem zu realisieren. Für den Bediener darf die Einbindung der Sensorik in den Bewegungsablauf keinen Zusatzaufwand bedeuten. Die einzustellenden Parameter, beispielsweise eine Maximalkraft, sollen Standard – Bewegungsparametern, wie Geschwindigkeit, gleichgestellt sein. In der Praxis ist aus einem individuell programmierten Bewegungsablauf die mitarbeitertypische Vorgehensweise, Erfahrung und Motivation erkennbar. Die aufgabenorientierte Schnittstelle soll durch weitestgehend festgelegte Abläufe sowie eingeschränkte Funktionsumfänge diese Einflussfaktoren minimieren. Dadurch erhöht sich gleichermaßen die Reproduzierbarkeit von Prüfungen.

Der Anspruch an die Schnittstelle zur Integration der Robotik besteht nicht darin, dass gänzlich unerfahrene Personen auf Anhieb mit der Technologie operieren können, wie es zunehmend im Bereich der Unterhaltungselektronik gelingt. Die Robotik ganz allgemein und die Anforderungen im Versuchswesen sind hierfür zu variabel. Mit Hilfe der Schnittstelle sollen Versuchingenieure, die unregelmäßig mit Robotikanwendungen in Berührung kommen, eigenständig und mit vertretbarem Zeitaufwand unkomplizierte Dauerversuche und Einzelmessungen durchführen können. Praxisnahe Beispiele sind die Messung eines Reibungskoeffizienten zweier Materialien zueinander, die Bestimmung der Steifigkeit eines Türblechs oder die in Abbildung 60 dargestellte Ermittlung des Kraftverlaufs beim Einclipsen eines Stifts in die Halterung.



**Abbildung 60: Messung des Kraftverlaufs beim Einclipsen eines Stifts in die Halterung**

Für künftige Versuchsanforderungen muss die Schnittstelle durch Softwareexperten um weitere Funktionen, insbesondere die Einbindung externer Sensoren, erweiterbar sein. Im Hinblick auf den Abgleich zwischen einer realen und virtuellen Erprobung sind Methoden gefragt, die eine Vergleichbarkeit der Ergebnisse beider Disziplinen optimieren.



### 4.6.2 Funktionsprinzip

Der hardwaretechnische Aufbau zur Anwendung der aufgabenorientierten Schnittstelle besteht aus dem Prüfstandsrechner, dem Robotercontroller und der externen Sensorik. Die Einheiten sind über Ethernet- und RS – 232 – Schnittstellen miteinander verbunden. Aufgrund der direkten Anbindung der externen Sensorik an den Robotercontroller ist eine unmittelbare Synchronisation der Sensordaten mit Positionswerten auf dem Robotercontroller möglich.

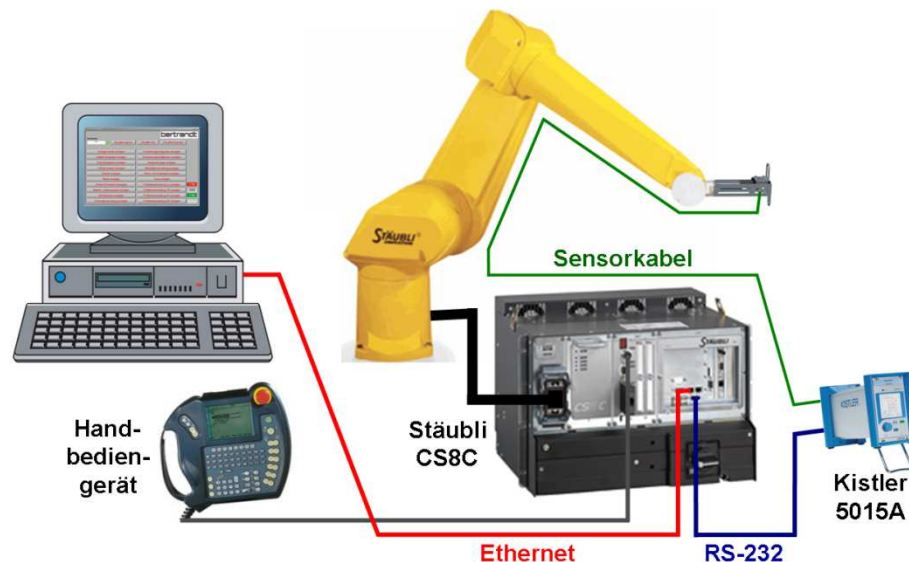


Abbildung 61: Hardwaretechnischer Aufbau der aufgabenorientierten Schnittstelle zur Robotik

Die Übertragung der Messwerte vom Ladungsverstärker *Kistler 5015A* zum Robotercontroller *Staubli CS8C* erfolgt im vorzeichenbehafteten Binär – Format. Zur Darstellung von vier signifikanten Stellen sind zwei Byte erforderlich. Für deren Übertragung werden zwei EIA – 232 – Datenrahmen benötigt, insgesamt ergeben sich mit eingerechnetem Overhead 22 Bit. Bei einer Rate von 1.000 Hertz sind so 22.000 Bit pro Sekunde zu übertragen. Beim Robotercontroller sowie am Ladungsverstärker lassen sich Übertragungsgeschwindigkeiten von bis zu 115.000 Bit pro Sekunde einstellen. Allerdings ist auf die Länge des Verbindungskabels zu achten, dies darf nicht länger als zwei Meter sein [Kis-07], [Bre-10b].

Den Kern der softwaretechnischen Bestandteile bildet das Kommunikationsprotokoll, welches die Botschaften und deren inhaltliche Zusammensetzung definiert. Zur Generierung der Botschaften dienen auf der Seite des Prüfsystems in LabVIEW geschriebene Funktionen, alternative Programmiersprachen mit TCP/IP – Unterstützung sind ebenso denkbar. Die Umsetzung der empfangenen Daten in Roboterbewegungen erfolgt mittels der herstellereigenen Programmierung auf dem Robotercontroller. Eine Bindung an ausgewählte Roboterhersteller besteht prinzipiell nicht, es müssen jedoch zur Umsetzung des Kommunikationsprotokolls bestimmte Funktionen seitens der Programmierumgebung des jeweiligen Herstellers zur Verfügung stehen. Dazu gehört beispielsweise das Auslesen der aktuellen Werkzeugposition. Eine Verfügbarkeit dieser Funktionen und die damit verbundene vollständige Anwendbarkeit des Kommunikationsprotokolls sind im Einzelfall zu prüfen.

Je nach Umfang des durch den Roboter auszuführenden Bewegungsablaufs gibt es zwei mögliche Varianten zur Inbetriebnahme durch den Bediener. Handelt es sich um *einfachste Betätigungen* im Rahmen einer Einzelmessung oder eines Dauerlaufs, beispielsweise das kraftgezielte Drücken eines Tasters, kann der Ablauf *online* mit Hilfe einer grafischen Bedienoberfläche und einem stark eingeschränkten Funktionsumfang durch den Versuchingenieur zu-

sammengestellt werden. Eine detaillierte Beschreibung zum Vorgang erfolgt in Abschnitt 4.6.5. *Komplexe Bewegungsabläufe*, wie das Öffnen und Schließen einer Fahrzeugtür, bestehen aus zusammengesetzten Bewegungen und sind zusätzlich von externen Sensoren abhängig. Derartige Abläufe basieren auf einem *Grundgerüst*, welches *offline* durch einen Softwareexperten entwickelt wird. Neben den Funktionen des Kommunikationsprotokolls stehen dazu auch Rechenoperationen der Vektorrechnung, beispielsweise *Winkel zwischen zwei Ebenen*, zur Verfügung. Zu jedem Grundgerüst bedarf es einer passenden Prozedur, durch die der Bediener *online* die Positionen und Parameter, beispielsweise Strecken oder Winkel, für den Bewegungsablauf festlegt. Dafür benötigte Steuerfunktionen, wie das Einlesen von Positionen oder die Abfrage von Parametern über das Handbediengerät, sind ebenfalls im Kommunikationsprotokoll enthalten. Aus der Kombination des Grundgerüsts einer Funktionsabfolge sowie der durch den Bediener vergebenen Parameter entsteht der komplexe Bewegungsablauf. Ein Beispiel für eine derartige Bewegungserstellung ist in Abschnitt 4.6.6 dokumentiert. Bei dieser zweitgenannten Variante ist während der Konzeption des Grundgerüsts seitens der Softwareexperten darauf zu achten, dass nur im Bedarfsfall feste Positionen im Raum vorzusehen sind. Besser ist die Verwendung von parameterbasierten Bewegungsfunktionen, so können durch Parameteränderungen manuelle oder automatisierte Anpassungen im Bewegungsablauf erfolgen. Ebenso werden Einflüsse durch mitarbeitertypische Vorgehensweisen beim Positionieren des Roboterwerkzeugs verringert. Die Vergabe eines Parameters, beispielsweise des Bewegungswinkels um eine Achse, ist mit wesentlich geringerem Zeitaufwand verbunden, als die Methode, drei Positionen zur Bildung einer Kreisbahn manuell mit dem Handbediengerät anzufahren und zu speichern.

Das Konzept des Kommunikationsprotokolls sieht vor, einen komplexen Bewegungsablauf in zusammenhängende Bewegungen aufzuteilen. Die Ausführung einer Bewegung startet erst, sobald sämtliche Informationen hierzu auf dem Robotercontroller vorliegen. Bei sensorabhängigen Bewegungen werden die Grenzwerte zusammen mit den Bewegungsparametern zum Robotercontroller übertragen, welcher dann die Überwachung und Bewegungssteuerung übernimmt. In beiden Fällen wird dadurch verhindert, dass bei einer verzögerten Datenübertragung, beispielsweise durch mangelnde *weiche* Echtzeit auf dem Prüfstandsrechner, stockende Bewegungen oder Grenzwertüberschreitungen auftreten.

Beide Varianten zur Erstellung eines Bewegungsablaufs setzen als Qualifikation im Bereich der Robotik voraus, dass der Bediener den Roboter manuell und unter textbasierter Anleitung über das Handbediengerät positionieren kann.

Diese aufgabenorientierte Variante eines hybriden Verfahrens zur Roboterprogrammierung nutzt zur genauen Positionierung des Roboterarms Bestandteile des Teach – In – Verfahrens. Dabei werden aber nur so wenige Positionen wie nötig fest vergeben und über das Handbediengerät eingelernt, die überwiegende Anzahl wird über Parameter berechnet. Eine kostenintensive Messtechnik ist daher nicht nötig. Individuelle Zusammenstellungen von *einfachsten Betätigungen* erfolgen *offline* am Prüfstandsrechner, allerdings nicht durch textbasierte Verfahren, sondern durch eine grafische Oberfläche mit bereichstypischen Bezeichnungen.

### 4.6.3 Kommunikationsprotokoll

Dieses Kapitel beschreibt die Botschaften für den Datenaustausch zwischen dem Prüfstandsrechner und dem Robotercontroller. Deren Inhalte sind so zusammengestellt, dass die Bewegungen des Roboterarms durch die Programmiervarianten *einfachste Betätigung* sowie *komplexer Bewegungsablauf* umsetzbar sind. Bestimmte Botschaften benötigen zusätzlich Berechnungen durch die Prüfsystemsoftware, diese werden im Zusammenhang erläutert.

In der Syntax einer Botschaft steht an erster Stelle ein Buchstabe, welcher die jeweilige Funktion festlegt. Nachrichten vom Prüfstandsrechner zum Robotercontroller starten mit Großbuchstaben, Antworten beginnen mit dem dazugehörigen Kleinbuchstaben. Dies ermöglicht bei einer Analyse des Datenverkehrs eine eindeutige Zuordnung. Es folgen die einzelnen Parameter durch ein Semikolon getrennt. Zahlen werden als vorzeichenbehafteter Integer übertragen, Kommazahlen müssen daher vor dem Senden mit einem der Genauigkeit entsprechenden Faktor multipliziert und beim Empfänger wieder dividiert werden. Eine Minimierung der zu übertragenden Bytes ist die Folge, vordergründiges Ziel ist aber die Vermeidung von länderspezifischen Trennzeichen in den als ASCII – Zeichenkette übertragenen Parametern. Jede Nachricht zum Robotercontroller endet mit einem Ausrufezeichen. Antworten zum Prüfsystem schließen mit dem Steuerzeichen <CR> für Wagenrücklauf. Tabelle 7 beschreibt die Syntax der Botschaften zum Abschalten der Armleistung.

<b>S</b>	<b>Syntax: U!</b>		
<b>E</b>	<b>Syntax: u[ArmL]&lt;CR&gt;</b>		
	[ArmL]	0	Armleistung ausgeschaltet
		1	Armleistung eingeschaltet

Tabelle 7: Beispielhafte Syntax des Kommunikationsprotokolls

Die mit einem **S** gekennzeichneten Zeilen beschreiben die zum Robotercontroller übertragene Befehle, das **E** steht für die empfangenen Daten der Antwortbotschaft. Direkt nach dem Dateneingang werden die entsprechenden Funktionen auf dem Robotercontroller in der herstellerspezifischen Programmierung ausgeführt. In diesem Beispiel führt dies zur Abschaltung der Armleistung. Die im Anschluss versendete Antwort dient zur Ausführungskontrolle und enthält in diesem Fall den aktualisierten Status der Armleistung.

Notwendige Berechnungen im Zusammenhang mit der Kommunikation, beispielsweise bei der Festlegung von Koordinatensystemen, werden seitens des Prüfsystems durch einen Lab-VIEW – basierten Programmcode ausgeführt. Der hierfür bestimmte Funktionsumfang ist im Vergleich zur Roboterprogrammiersprache wesentlich ausgedehnter.

#### 4.6.3.1 Steuerbotschaften

Steuerbotschaften bezwecken, im Robotercontroller gespeicherte Parameter, beispielsweise Punkte zur Festlegung von Koordinatensystemen, einzustellen, ebenso können Zustände, wie die Betriebsart, ausgelesen werden. Zusätzliche Bestandteile sind die Steuerung des Messdatenexports, der Bewegungsfreigabe sowie der Bedienerkommunikation. Der Anwender hat keinen direkten Zugriff auf die Steuerbotschaften. Deren Verwendung geschieht nur bei der Erstellung der Grundgerüste für Bewegungsabläufe durch Softwareexperten.

##### 4.6.3.1.1 Position auslesen

Die Botschaft *Position auslesen* liefert die gegenwärtige Position des eingestellten Roboterwerkzeugs im ausgewählten Koordinatensystem.

S)	<b>Syntax: P[KS]!</b>		
	[KS]	0	Roboterkoordinatensystem
		1	Bauteilkoordinatensystem
		2	Türkoordinatensystem
E)	[X]	3	Fahrzeugkoordinatensystem
		<b>Syntax: p[X];[Y];[Z];[Rx];[Ry];[Rz];[Status]&lt;CR&gt;</b>	
		[X] X – Koordinate im ausgewählten KS [mm ×100]	
		[Y] Y – Koordinate im ausgewählten KS [mm ×100]	
		[Z] Z – Koordinate im ausgewählten KS [mm ×100]	
		[Rx] Rotation um die X – Achse im ausgewählten KS [° ×100]	
		[Ry] Rotation um die Y – Achse im ausgewählten KS [° ×100]	
		[Rz] Rotation um die Z – Achse im ausgewählten KS [° ×100]	
	[Status]	0	kein Fehler
		1	fehlerhafte Auswahl des Parameters [KS]

Tabelle 8: Syntax der Botschaft *Position auslesen*

Die in der Antwort empfangenen Daten dienen als Eingangsgrößen für die Botschaft *Position anfahren* oder für Berechnungen zum Festlegen von Koordinatensystemen.

#### 4.6.3.1.2 Bedienerkommunikation

Mit Hilfe der Botschaft *Bedienerkommunikation* erfolgt die Ausgabe von Hinweisen oder Parameterabfragen auf dem Display des Handbediengeräts. Dadurch ist es möglich, den Bediener durch Informationen und Anweisungen bei der Positionierung des Roboterarms zu unterstützen.

S)	Syntax: I[TextIndex];[StdWert]!		
	[TextIndex]	Index des auf dem Display am Handbediengerät anzuzeigenden Texts	
	[StdWert]	voreingestellter Wert einer Parameterabfrage	
E	Syntax: i[Wert];[Status]<CR>		
	[Wert]	am Handbediengerät eingegebener Parameterwert	
	[Status]	0	kein Fehler
		1	Wert des Parameters <i>[TextIndex]</i> nicht definiert

Tabelle 9: Syntax der Botschaft *Bedienerkommunikation*

Der in Abhängigkeit des Parameters [TextIndex] auszugebende Text muss im Programmcode auf dem Robotercontroller hinterlegt sein. Eine Antwortbotschaft wird erst übertragen, sobald der Bediener den Hinweis quittiert oder Parameter eingegeben hat. Der Datentyp für die Parameter [StdWert] und [Wert] ist als vorzeichenbehafteter Integer festgelegt. Bei Textausgaben ohne Eingabeaufforderung am Handbediengerät wird der Parameter [StdWert] ignoriert, der Inhalt von [Wert] in der Antwort bekommt eine 0 zugewiesen.

#### 4.6.3.1.3 Koordinatensystem festlegen

Bewegungsabläufe, deren Steuerung auf Basis eines Grundgerüsts erfolgt, benötigen grundsätzlich ein während der Inbetriebnahme einzumessendes Koordinatensystem. Je nach Prü-

fung und primär verwendeten Bewegungsformen eignet sich das Bauteil-, Tür- oder Fahrzeugkoordinatensystem.

Das Bauteilkoordinatensystem ist prädestiniert für Tests, bei denen identische Bauteile geprüft werden oder regelmäßig Umbauten, beispielsweise in eine Klimakammer, stattfinden. Dabei wird die Ausrichtung des Prüflings zum Roboter festgelegt. Das Türkoordinatensystem ist für Prüfungen konzipiert, bei denen die Hauptbewegungen um eine feste Drehachse rotieren. Die Drehachse des Bauteils stimmt in diesem Fall mit der X – Achse des Türkoordinatensystems überein. Das Fahrzeugkoordinatensystem definiert allgemein den Nullpunkt eines Fahrzeugs und ist besonders bei CAD – Modellen als Referenz unverzichtbar. Eine Synchronisation des Fahrzeugkoordinatensystems mit dem Roboterkoordinatensystem ermöglicht es, CAD – basierte Simulationen für Vergleiche exakt in der Realität nachzustellen. Durch die Festlegung von Koordinatensystemen wird erreicht, dass der Roboterarm und die Prüflinge immer in gleicher Weise zueinander positioniert sind. Genauere Vergleiche sowie eine bessere Reproduzierbarkeit sind das Ergebnis.

Jede der drei koordinatensystemspezifischen Methoden zur Festlegung beginnt damit, dass der Bediener mit einer am Roboterflansch montierten Messspitze bis zu vier Punkte,  $\overline{P}_1$  bis  $\overline{P}_4$ , möglichst genau mit dem Handbediengerät anfährt. Die Koordinaten der Punkte werden mittels der Funktion *Position auslesen* erfasst und als Basis für die Berechnung der Lage des festzulegenden Koordinatensystems verwendet. Das Ergebnis der Berechnung sind wiederum drei Punkte, welche die Lage des Koordinatensystems im Raum eindeutig definieren.  $\overline{P}_0$  ist der Koordinatenursprung,  $\overline{P}_X$  ist ein Punkt auf der X – Achse und  $\overline{P}_{XY}$  ist ein Punkt in der X – Y – Ebene.



Abbildung 62: Festlegen des Türkoordinatensystems

Nach der vom Koordinatensystem abhängigen Berechnung durch eine der nachfolgend beschriebenen Methoden werden die Lagen der Punkte  $\overline{P}_0$ ,  $\overline{P}_X$  und  $\overline{P}_{XY}$  zum Robotercontroller übertragen.

S)

Syntax: G[KS];[P0_X];[P0_Y];[P0_Z];[PX_X];[PX_Y];[PX_Z];[PXY_X];[PXY_Y];[PXY_Z]!		
[KS]	0	Bauteilkoordinatensystem
	1	Türkoordinatensystem
	2	Fahrzeugkoordinatensystem
[P0_X]		X – Koordinate des Punkts P0 im Welt – KS des Roboters [mm ×100]
[P0_Y]		Y – Koordinate des Punkts P0 im Welt – KS des Roboters [mm ×100]
⋮		⋮
[PXY_Z]		Z – Koordinate des Punkts PXY im Welt – KS des Roboters [mm ×100]

E)

Syntax: g[Status]<CR>		
[Status]	0	kein Fehler
	1	fehlerhafte Auswahl des Parameters [KS]
	2	fehlerhafte Festlegung des Koordinatensystems, ein Punkt liegt zu nah am Ursprung

Tabelle 10: Syntax der Botschaft *Koordinatensystem festlegen***Variante 1: Lage des Bauteilkoordinatensystems übergeben**

$\vec{P}_1$ ,  $\vec{P}_2$  und  $\vec{P}_3$  werden zum Festlegen des Bauteilkoordinatensystems ohne eine Umrechnung direkt an die Botschaft *Koordinatensystem festlegen* als  $\vec{P}_0$ ,  $\vec{P}_x$  und  $\vec{P}_{xy}$  übergeben. Der Bediener muss beim Vermessen der Punkte darauf achten, dass diese möglichst große Abstände zueinander aufweisen und weit entfernt von einer gemeinsamen Gerade liegen. Dadurch verringert sich der Einfluss von Ungenauigkeiten beim manuellen Positionieren der Messspitze. Allgemein ist zu beachten, dass die Einmesspunkte an sehr steifen Orten mit geringen Toleranzen liegen.

**Variante 2: Lage des Türkoordinatensystems berechnen**

Rotiert ein Bewegungsablauf mehrfach um eine feste Drehachse, wie beim Öffnen und Schließen einer Fahrzeughür, ist die Festlegung der Drehachse über ein Türkoordinatensystem zweckmäßig. Dafür muss derselbe Punkt am rotierenden Bauteil an drei verschiedenen Winkelstellungen mit der Messspitze angefahren werden. Auf Basis der so erhaltenen Punkte  $\vec{P}_1$ ,  $\vec{P}_2$  und  $\vec{P}_3$  ist eine Berechnung der Drehachse möglich. Der Richtungsvektor  $\vec{a}$  der Drehachse wird dafür aus dem Kreuzprodukt der durch die Punkte verlaufenden Geraden gebildet.

$$\vec{a} = (\vec{P}_2 - \vec{P}_1) \times (\vec{P}_3 - \vec{P}_1)$$

**Formel 3: Richtungsvektor der Bauteildrehachse**

Mit Hilfe der zwei Formeln

$$\vec{g}_1(\lambda_1) = \left( \vec{P}_1 + \frac{1}{2}(\vec{P}_2 - \vec{P}_1) \right) + \lambda_1 \left( (\vec{P}_2 - \vec{P}_1) \times \vec{a} \right)$$

**Formel 4: Gerade 1 durch die Drehachse des Türkoordinatensystems**

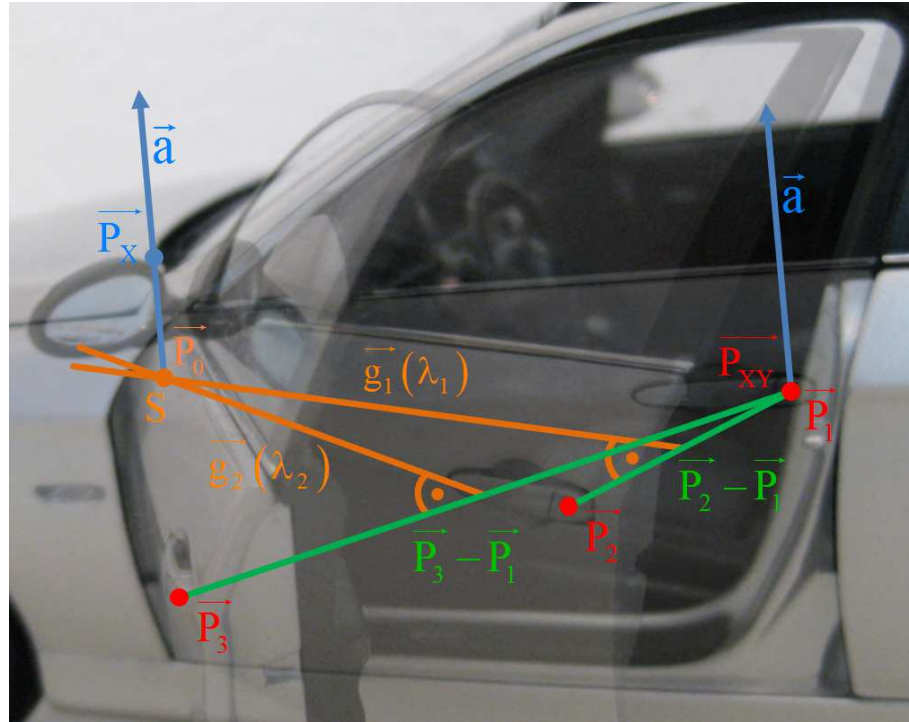
und



$$\vec{g}_2(\lambda_2) = \left( \vec{P}_1 + \frac{1}{2}(\vec{P}_3 - \vec{P}_1) \right) + \lambda_2 \left( (\vec{P}_3 - \vec{P}_1) \times \vec{a} \right)$$

**Formel 5: Gerade 2 durch die Drehachse des Türkoordinatensystems**

sind zwei Geraden beschrieben, welche sich genau in einem Punkt S auf der Drehachse schneiden.



**Abbildung 63: Punkte, Vektoren und Geraden zur Berechnung des Fahrzeugkoordinatensystems**

Durch Gleichsetzen von  $\vec{g}_1(\lambda_1)$  und  $\vec{g}_2(\lambda_2)$  in der Komponentenschreibweise können die Unbekannten  $\lambda_1$  oder  $\lambda_2$  mit Hilfe eines linearen Gleichungssystems berechnet werden.

$$\lambda_2 = \frac{B_y(A_x - C_x) + B_x(C_y - A_y)}{B_y \cdot D_x - B_x \cdot D_y}$$

**Formel 6: Lambda 2 der Geradengleichung**

mit

$$\vec{A} = \vec{P}_1 + \frac{1}{2}(\vec{P}_2 - \vec{P}_1), \vec{B} = (\vec{P}_2 - \vec{P}_1) \times \vec{a}, \vec{C} = \vec{P}_1 + \frac{1}{2}(\vec{P}_3 - \vec{P}_1), \vec{D} = (\vec{P}_3 - \vec{P}_1) \times \vec{a}$$

Der durch Einsetzen von  $\lambda_2$  in Formel 5 erhaltene Punkt ist die Lage des Ursprungs  $\vec{P}_0$  des Türkoordinatensystems.  $\vec{P}_x$  befindet sich ebenfalls auf der Drehachse in einem positiven Abstand von 100 Millimetern zum Ursprung.  $\vec{P}_{xy}$  ist gleich dem ersten mit der Messspitze angefahrenen Punkt  $\vec{P}_1$ , welcher in einer definierten Bauteilstellung, beispielsweise bei geschlossener Tür, eingemessen wird. Abbildung 64 zeigt die programmtechnische Umsetzung der Formeln 3 bis 6 in LabVIEW.



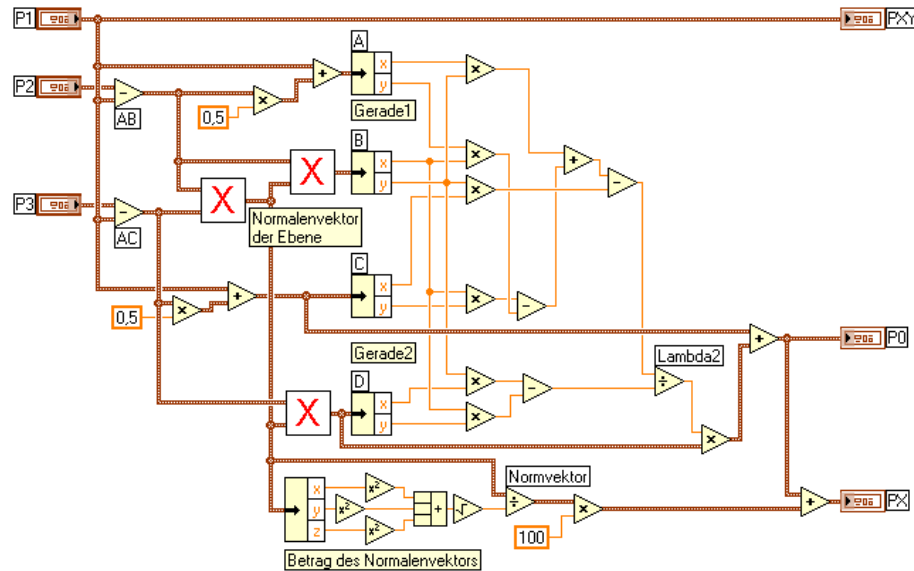


Abbildung 64: Programmcode zur Berechnung der Lage der Punkte des Türkoordinatensystems

### Variante 3: Lage des Fahrzeugkoordinatensystems berechnen

Für Steifigkeitsmessungen an Fahrzeugkomponenten sind die Koordinaten der Kraftangriffspunkte sowie die Ausrichtung des Prüfstempels als Vektor im Lastenheft definiert. Um die Vergleichbarkeit zwischen den virtuellen und realen Messungen zu optimieren, ist eine Synchronisation des Fahrzeug- und Roboterkoordinatensystems zweckmäßig. Anschließend können Steifigkeitsmessungen über den Roboter automatisiert auf Basis der Daten aus dem Lastenheft durchgeführt werden. Bei dieser Methode entfällt das vorherige Markieren der Prüfpunkte am Bauteil durch externe Einmessverfahren.

Zur Synchronisation der Koordinatensysteme sind zunächst vier eindeutige Punkte  $\overrightarrow{P_{1,r}}$  bis  $\overrightarrow{P_{4,r}}$  am realen Bauteil mit Hilfe der Messspitze anzufahren und einzulesen. Diese Punkte dürfen nicht in einer gemeinsamen Ebene liegen, sie müssen zur Minimierung von Ungenauigkeiten beim Positionieren der Messspitze eindeutig definierbar sein und einen möglichst großen Abstand zueinander aufweisen. Weiterhin sind die Koordinaten der entsprechenden Punkte  $\overrightarrow{P_{1,v}}$  bis  $\overrightarrow{P_{4,v}}$  mittels einer CAD – Software aus dem virtuellen Modell des Fahrzeugs zu exportieren. Anhand dieser vier Punktepaare ist ein Abgleich möglich. Für die Festlegung des Fahrzeugkoordinatensystems ist zu berechnen, wie die zu prüfende Fahrzeugkomponente mit dem dazugehörigen Koordinatensystem im Roboterkoordinatensystem positioniert ist. Dies ist mit Hilfe der linearen Abbildungsgleichungen möglich.

$$P_{1,r_x} = a_{11} \cdot P_{1,v_x} + a_{12} \cdot P_{1,v_y} + a_{13} \cdot P_{1,v_z} + f_1$$

$$P_{1,r_y} = a_{21} \cdot P_{1,v_x} + a_{22} \cdot P_{1,v_y} + a_{23} \cdot P_{1,v_z} + f_2$$

$$P_{1,r_z} = a_{31} \cdot P_{1,v_x} + a_{32} \cdot P_{1,v_y} + a_{33} \cdot P_{1,v_z} + f_3$$

$$P_{2,r_x} = a_{11} \cdot P_{2,v_x} + a_{12} \cdot P_{2,v_y} + a_{13} \cdot P_{2,v_z} + f_1$$

⋮

$$P_{4,r_z} = a_{31} \cdot P_{4,v_x} + a_{32} \cdot P_{4,v_y} + a_{33} \cdot P_{4,v_z} + f_3$$

**Formel 7: Lineare Abbildungsgleichungen**

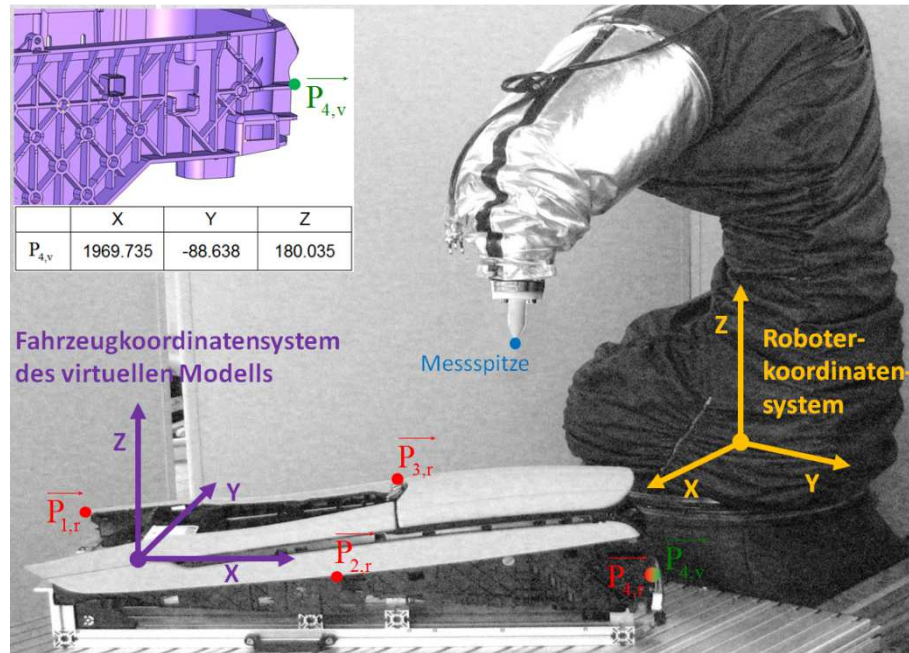


Abbildung 65: Punkte zur Berechnung der Lage des Fahrzeugkoordinatensystems

Durch diese zwölf Abbildungsgleichungen lassen sich die zwölf Unbekannten der Transformationsmatrix  $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$  und des Vektors  $\vec{f} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$  bestimmen. In der programmtechnischen Umsetzung dient dazu das Determinantenverfahren.

Mit Hilfe der Formel

$$\vec{P}_{\text{RobKS}} = A \cdot \vec{P}_{\text{FzgKS}} + \vec{f}$$

**Formel 8: Lineare Abbildung zur Synchronisation von Koordinatensystemen**

können dann die Koordinaten eines Punkts  $\vec{P}_{\text{FzgKS}}$  im Fahrzeugkoordinatensystem in die Koordinaten eines Punkts  $\vec{P}_{\text{RobKS}}$  im Roboterkoordinatensystem umgerechnet werden. Kon-

kret wird diese Berechnung für die Punkte  $\vec{P}_{0_{\text{FzgKS}}} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ ,  $\vec{P}_{x_{\text{FzgKS}}} = \begin{pmatrix} 100\text{mm} \\ 0 \\ 0 \end{pmatrix}$  und

$\vec{P}_{xy_{\text{FzgKS}}} = \begin{pmatrix} 0 \\ 100\text{mm} \\ 0 \end{pmatrix}$  durchgeführt. Die Ergebnisse für  $\vec{P}_0$ ,  $\vec{P}_x$  sowie  $\vec{P}_{xy}$  dieser drei Berech-

nungen werden über die Botschaft *Koordinatensystem festlegen* an den Roboter übertragen.

Abbildung 66 beinhaltet den LabVIEW – Programmcode zur Anwendung der Formeln 7 und 8 sowie des Determinantenverfahrens. Linksseitig sind die vier mittels der Messspitze eingelesenen und die vier aus dem CAD – Modell ermittelten Punkte positioniert. Zur Berechnung der Koeffizientendeterminante werden die einzelnen Komponenten der Punkte des CAD – Modells durch drei verschachtelte *For – Schleifen* in eine Matrix überführt. Die anschließende *For – Schleife* erstellt die Hilfsdeterminanten durch Einsetzen der eingemessenen Punkte mit anschließender Berechnung der unbekannten Koeffizienten der Matrix A und des Vektors f.

Der letzte Schritt ist die Umrechnung eines Punkts aus dem Fahrzeugkoordinatensystem mit Hilfe einer Matrizenmultiplikation und Addition entsprechend der Formel 8 ins Roboterkoordinatensystem.

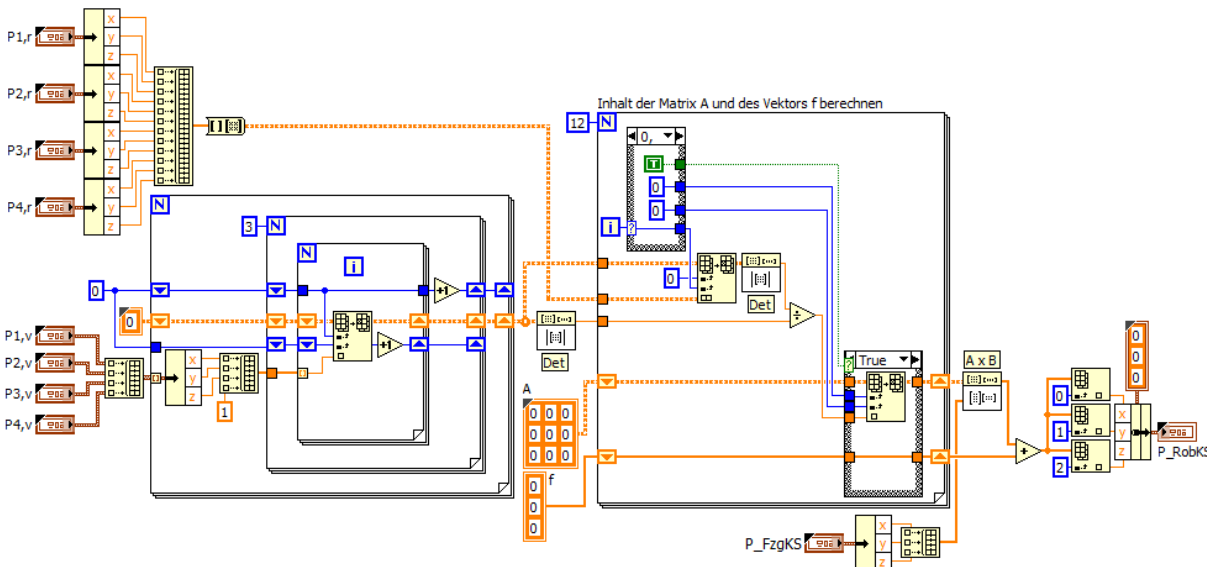


Abbildung 66: Programmcode zur Umrechnung vom Bauteil- ins Roboterkoordinatensystem

4.6.3.1.4 Überschleifen

Die Botschaft *Überschleifen* legt die Parameter der gleichnamigen Funktion auf dem Robotercontroller fest. Diese Funktion erlaubt ein Verlassen der Bahnkurve in gewissen Grenzen, um Zeit sowie Energie zu sparen und die Mechanik des Roboterarms zu schonen.

S)	Syntax: S[Aktiv];[Verlassen];[Erreichen]!		
	[Aktiv]	0	Überschleifen deaktivieren
		1	Überschleifen aktivieren
	[Verlassen]	Entfernung vor der Zielposition beim Start der Glättung	
	[Erreichen]	Entfernung hinter der Zielposition beim Ende der Glättung	
E	Syntax: s[Status]<CR>		
	[Status]	0	kein Fehler
		1	fehlerhafte Auswahl des Parameters <i>[Aktiv]</i>

Tabelle 11: Syntax der Botschaft *Überschleifen*

Bei einer Deaktivierung der Funktion durch die Übertragung des Werts 0 im Parameter [Aktiv] werden die Inhalte von [Verlassen] und [Erreichen] ignoriert.

4.6.3.1.5 Bewegungsfreigabe

Über die Botschaft *Bewegungsfreigabe* wird die Ausführung der im Speicher des Robotercontrollers hinterlegten Bewegungsabläufe gesteuert. Während eines Bewegungsablaufs mit parallelem Messdatenexport ist aus Gründen der Performance die zeitgleiche Verarbeitung von Botschaften durch den Robotercontroller zu vermeiden. Dennoch müssen für den effektiven Einsatz der Überschleif – Funktion zum Glätten der Bahnkurve deren einzelne Abschnitte im Robotercontroller vorliegen. Um dies zu gewährleisten, wird die Bewegungsfreigabe zu-

nächst entzogen. Die anschließende Übertragung zusammenhängender Bewegungsbotschaften bewirkt die Generierung der Bahnkurve in Abhängigkeit der Parameter *[Aktiv]*, *[Verlassen]* und *[Erreichen]*. Im Anschluss erfolgt die Bewegungsfreigabe, bis der Roboter das Ziel des aktuellen Bewegungsablaufs erreicht hat.

S>	<b>Syntax: C[Art]!</b>		
	[Art]	0	Bewegungsfreigabe erteilen
		1	Bewegungsfreigabe sofort entziehen
		2	Bewegungsfreigabe nach Bewegungsende entziehen
		3	Bewegungsfreigabe erteilen und nach Bewegungsende entziehen
E<	<b>Syntax: c[Status]&lt;CR&gt;</b>		
	[Status]	0	kein Fehler
		1	fehlerhafte Auswahl des Parameters <i>[Art]</i>

Tabelle 12: Syntax der Botschaft *Bewegungsfreigabe*

Die Antwortbotschaft wird bei den Werten 2 und 3 für den Parameter *[Art]* erst übertragen, nachdem die Roboterbewegung abgeschlossen ist.

#### 4.6.3.1.6 Messmittel- und Werkzeugparameter

##### Ladungsverstärker parametrisieren

Diese Botschaft dient zur Parametrierung der Messhardware. Im folgenden Beispiel werden die Einstellungen für den Ladungsverstärker eines piezoelektrischen Kraftsensors übermittelt, der über eine RS – 232 Schnittstelle direkt mit dem Robotercontroller verbunden ist. Für einen Austausch der Kraftmessdose oder bei prüfungsbedingten Anpassungen des Messbereichs ist eine Änderbarkeit der Parameter notwendig. Diese werden vom Robotercontroller an den Ladungsverstärker weitergeleitet. Zur Adaption weiterer Sensortypen ist eine Anpassung des Kommunikationsprotokolls erforderlich.

S>	Syntax: B[Aktion];[Wert]!		
	[Aktion]	0	Parameter des Ladungsverstärkers abfragen
		1	Messbereich des Kraftsensors einstellen
		2	Empfindlichkeit des Kraftsensors einstellen
	[Wert]	Zahlenwert des einzustellenden Parameters [Einheit ×100]	
E<	Syntax: b[Messb];[Empf];[Status]<CR>		
	[Messb]	eingestellter Messbereich im Ladungsverstärker [Einheit ×100]	
	[Empf]	eingestellte Empfindlichkeit im Ladungsverstärker [Einheit ×100]	
	[Status]	0	kein Fehler
		1	keine Rückmeldung vom Ladungsverstärker
		2	fehlerhafte Auswahl des Parameters [Aktion]

Tabelle 13: Syntax der Botschaft *Ladungsverstärker parametrisieren*

Beim Abfragen der Parameter durch Übertragung des Werts 0 im Feld *[Aktion]* wird der Parameter *[Wert]* ignoriert. Nach jeder Änderung von Parametern werden diese aus der angeschlossenen Messhardware zurückgelesen und zur Kontrolle in der Antwortbotschaft übertragen.

### Werkzeugparameter einstellen

Über die Botschaft *Werkzeugparameter einstellen* werden die Ausmaße und Verdrehungen des aktuell verwendeten Roboterwerkzeugs im Robotercontroller festgelegt. Die Möglichkeit zur automatischen Änderung ist notwendig, da das am Roboterflansch montierte Werkzeug mehrere Einzelwerkzeuge beinhalten kann. Beispiele werden in Abschnitt 5.1.2 beschrieben.

S)	<b>Syntax:</b> T[X];[Y];[Z];[Rx];[Ry];[Rz];[KraftFaktor]!	
	[X]	Ausmaß in X – Richtung [mm ×100]
	[Y]	Ausmaß in Y – Richtung [mm ×100]
	[Z]	Ausmaß in Z – Richtung [mm ×100]
	[Rx]	Rotation um die X – Achse des Falsch – KS [° ×100]
	[Ry]	Rotation um die Y – Achse des Falsch – KS [° ×100]
	[Rz]	Rotation um die Z – Achse des Falsch – KS [° ×100]
	[KraftFaktor]	Umrechnungsfaktor des gemessenen Kraftwerts
E	<b>Syntax:</b> t[Status]<CR>	
	[Status]	0 keine Änderung der Werkzeugparameter
		1 Änderung der Werkzeugparameter

Tabelle 14: Syntax der Botschaft *Werkzeugparameter einstellen*

Wird die Kraft am Sensor nicht direkt, sondern über einen Hebelarm eingeleitet, muss der Umrechnungsfaktor über den Parameter *[KraftFaktor]* eingestellt werden. Die Umrechnung erfolgt auf dem Robotercontroller, da für überlastungsbedingte Sicherheitsabschaltungen die tatsächliche Belastung auf den Roboterarm ausschlaggebend ist.

Der Parameter *[Status]* in der Antwortbotschaft zeigt an, ob sich die Werkzeugparameter mit der aktuellen Botschaft verändert haben. Diese Information soll versehentliche Änderungen verhindern, da eine Modifikation der Werkzeugparameter eine Auswirkung auf die Roboterbewegung hat.

#### 4.6.3.1.7 Messdatenexport

Mit dieser Botschaft wird der kontinuierliche Export von Messdaten gesteuert.

S)	<b>Syntax:</b> H[Aktion]!	
	[Aktion]	0 Messdatenexport deaktivieren
		1 Messdatenexport aktivieren
E	<b>Syntax:</b> h[Status]<CR>	
	[Status]	0 Messdatenexport nicht aktiv
		1 Messdatenexport aktiv
		2 fehlerhafte Auswahl des Parameters <i>[Aktion]</i>

Tabelle 15: Syntax der Botschaft *Messdatenexport*

Bei aktiviertem Messdatenexport wird bis zur Deaktivierung kontinuierlich die nachfolgende Botschaft übertragen.

<b>E</b>	<b>Syntax: d[Kraft];[Beweg];[X];[Y];[Z]&lt;CR&gt;</b>	
	[Kraft]	Messwert der Kraft [N × 10]
	[Beweg]	0 Linearbewegung
		1 Hinfahrt der Kreisbewegung
		2 Rückfahrt der Kreisbewegung
	[X]	X – Koordinate im Roboterkoordinatensystem [mm × 100]
	[Y]	Y – Koordinate im Roboterkoordinatensystem [mm × 100]
	[Z]	Z – Koordinate im Roboterkoordinatensystem [mm × 100]

Tabelle 16: Syntax der Botschaft zur Übertragung der Messdaten

Der Parameter *[Beweg]* wird verwendet, um bei Kreisbewegungen aus den einzelnen empfangenen Koordinaten die zurückgelegte Strecke für Kraft – Weg – Diagramme zu ermitteln. Bei der *Hinfahrt* erfolgt eine Addition der Wegänderung zur bereits zurückgelegten Strecke, bei der *Rückfahrt* eine Subtraktion. Für Linearbewegungen ist der Parameter *[Beweg]* nicht relevant, da sich die Strecke aus der Differenz von aktueller Position und Startposition ergibt.

Die Leistungsfähigkeit des Robotercontrollers ist entscheidend für die Übertragungsrate der Messdaten. Bei ausgiebigen Tests mit einem Robotercontroller vom Typ *CS8C* der Firma *Stäubli* konnten 500 Hertz erreicht werden.

Je nach verwendeter zusätzlicher Messhardware kann der Inhalt der Botschaft veränderte oder zusätzliche Parameter erfordern. Für diesen Zweck ist eine Anpassung des Kommunikationsprotokolls durch Softwareexperten unerlässlich.

#### 4.6.3.1.8 Achsmomentexport

Die Botschaft *Achsmomentexport* steuert die kontinuierliche Übertragung der Drehmomente der Roboterachsen. Die Botschaft findet bei Steifigkeitsmessungen Verwendung, um die Zuverlässigkeit der Drehmomente an den einzelnen Roboterachsen zu überwachen. Steifigkeitsmessungen erfolgen quasistatisch, daher ist eine Übertragungsrate von 10 Hertz ausreichend.

<b>S</b>	<b>Syntax: Q[Aktion]!</b>	
	[Aktion]	0 Achsmomentexport deaktivieren
		1 Achsmomentexport aktivieren
<b>E</b>	<b>Syntax: q[Status]&lt;CR&gt;</b>	
	[Status]	0 Achsmomentexport nicht aktiv
		1 Achsmomentexport aktiv
		2 fehlerhafte Auswahl des Parameters <i>[Aktion]</i>

Tabelle 17: Syntax der Botschaft *Achsmomentexport*

Bei aktiviertem Achsmomentexport wird bis zur Deaktivierung kontinuierlich die folgende Botschaft übertragen.

<b>E</b>	<b>Syntax: x[AMJ1];[AMJ2];[AMJ3];[AMJ4];[AMJ5];[AMJ6]&lt;CR&gt;</b>	
	[AMJ1]	Achsmoment der Roboterachse 1 [Nm]
	⋮	⋮
	[AMJ6]	Achsmoment der Roboterachse 6 [Nm]

Tabelle 18: Syntax der Botschaft zur Übertragung der Achsmomente



Die übertragenen Werte sind lediglich ein Maß für die Größenordnung der Momente, beeinflusst durch Schwerkraft, Reibung, Massenträgheit und Genauigkeit der Stromsensoren. Auf Basis dieser Daten können die extern einwirkenden Kräfte jedoch nicht mit einer für das Versuchswesen typischen Genauigkeit berechnet werden.

#### 4.6.3.1.9 Armleistung und Betriebsart

##### Status Armleistung und Betriebsart

Diese Botschaft übermittelt die aktuell eingestellte Betriebsart des Roboters sowie den Status der Armleistung. Die Instruktion des Bedieners zur Parametrierung von Grundgerüsten komplexer Bewegungsabläufe erfolgt über das Handbediengerät. Dabei sind mehrmals Wechsel der Betriebsart oder Schaltungen der Armleistung vorzunehmen. Mit Hilfe dieser Botschaft werden die Eingaben der Bediener kontrolliert.

S)	Syntax: A!		
E	Syntax: a[ArmL];[BArt]<CR>		
	[ArmL]	0	Armleistung ausgeschaltet
		1	Armleistung eingeschaltet
	[BArt]	0	ungültig oder Betriebsartwechsel
		1	Handbetrieb
		2	Testmodus
		3	lokale Automatik
		4	ferngesteuerte Automatik

Tabelle 19: Syntax der Botschaft *Status Armleistung und Betriebsart*

Die Antwortbotschaft *Status Armleistung und Betriebsart* wird auf Anfrage oder ereignisgesteuert beim Wechsel der Parameter [ArmL] oder [BArt] automatisch vom Robotercontroller übertragen.

##### Armleistung ausschalten

Die Botschaft *Armleistung ausschalten* führt zur sofortigen Deaktivierung der Armleistung.

S)	Syntax: U!		
E	Syntax: u[ArmL]<CR>		
	[ArmL]	0	Armleistung ausgeschaltet
		1	Armleistung eingeschaltet

Tabelle 20: Syntax der Botschaft *Armleistung ausschalten*

Nach Abschluss der Prüfungen oder bei fehlerbedingten Stillständen ist es sinnvoll, die Armleistung aus Sicherheitsgründen und zur Maschinenschonung automatisiert abzuschalten. Ein automatisches Einschalten der Armleistung ist nicht möglich.



### 4.6.3.2 Bewegungsbotschaften

Mit Ausnahme der Botschaft *Position anfahren*, die als Eingangsparameter eine zuvor ausgelesene oder berechnete Position benötigt, beinhalten die übrigen Botschaften zur Generierung von Bewegungen flexibel einstellbare Parameter. Dabei wird die Roboterbewegung durch Strecken sowie Winkel definiert und auf Basis einer *fiktiven Werkzeugposition* berechnet. Wie bereits in Abschnitt 4.6.3.1.5 erläutert, erfolgt aus Performancegründen eine gesammelte Übertragung zusammenhängender Bewegungsbotschaften. Die fiktive Werkzeugposition ist in diesem Kontext die Zielposition der vorherigen Bewegungsbotschaft, die als Startposition für die nachfolgende Bewegungsbotschaft dient. Die fiktive Werkzeugposition wird als Variable *pBasis* im Roboterkoordinatensystem auf dem Controller gespeichert und nach jeder Verarbeitung einer Bewegungsbotschaft aktualisiert. Die Antwortbotschaften liefern die Zielpositionen des Roboterwerkzeugs im ausgewählten Koordinatensystem, hiermit lassen sich vektorielle Berechnungen aus dem Anwendungsgebiet der Geometrie in das Grundgerüst eines *komplexen Bewegungsablaufs* mit einbeziehen. Parameterabfragen und Eingaben durch den Bediener werden dadurch reduziert.

Durch diesen parameterbasierten Ansatz ergeben sich wichtige Funktionalitäten für die Programmierung typischer Bewegungsabläufe im Versuchswesen: Bei Steifigkeitsprüfungen liegt die Zielposition inmitten des Bauteils und kann nicht über die Teach – In – Methode angefahren werden. Durch die parameterbasierte Linearbewegung wird die Prüfbewegung möglich. Die Bewegungen lassen sich durch Parameter manuell oder automatisiert beeinflussen, ohne dass eine Unterbrechung im Testablauf nötig ist. Weiterhin hat sich in der Praxis das Teach von Kreisbewegungen durch drei Positionen unter Kraftschluss oder bei kleinen Radien als sehr schwierig erwiesen, beispielsweise beim Entriegeln eines Türgriffs oder beim Betätigen eines Gangwahlhebels. Für diesen Zweck stehen nun alternative Methoden zur Verfügung. Das Messen von Strecken und Winkeln ist im Gegensatz zum manuellen Anfahren von Positionen mit weniger Zeitaufwand verbunden. Zudem fallen die mitarbeitertypischen Einflüsse beim Messen geringer ins Gewicht als beim Positionieren des Roboterarms.

#### 4.6.3.2.1 Position anfahren

Die Botschaft *Position anfahren* generiert eine Bewegung des Roboterwerkzeugs zur angegebenen Position im ausgewählten Koordinatensystem.

S)	<b>Syntax: M[X];[Y];[Z];[Rx];[Ry];[Rz];[KS];[OffsetZ];[ZP_KS];[Geschw];[Beschl];[Bremsen]!</b>	
	[X]	X – Koordinate der Zielposition im ausgewählten KS [mm ×100]
	[Y]	Y – Koordinate der Zielposition im ausgewählten KS [mm ×100]
	[Z]	Z – Koordinate der Zielposition im ausgewählten KS [mm ×100]
	[Rx]	Rotation um die X – Achse im ausgewählten KS [° ×100]
	[Ry]	Rotation um die Y – Achse im ausgewählten KS [° ×100]
	[Rz]	Rotation um die Z – Achse im ausgewählten KS [° ×100]
	[KS]	0 Roboterkoordinatensystem
		1 Bauteilkoordinatensystem
		2 Türkoordinatensystem
		3 Fahrzeugkoordinatensystem
	[OffsetZ]	Offset in Stoßrichtung des Werkzeugs [mm ×100]

	[ZP_KS]	0	Ausgabe der Zielposition im Roboterkoordinatensystem
		1	Ausgabe der Zielposition im Bauteilkoordinatensystem
		2	Ausgabe der Zielposition im Türkoordinatensystem
		3	Ausgabe der Zielposition im Fahrzeugkoordinatensystem
	[Geschw]	Geschwindigkeit der Roboterbewegung [%]	
	[Beschl]	positive Beschleunigung der Roboterbewegung [%]	
	[Bremsen]	negative Beschleunigung der Roboterbewegung [%]	
⌈E	Syntax: m[ZP_X];[ZP_Y];[ZP_Z];[Status]<CR>		
	[ZP_X]	X – Koordinate der Zielposition im ausgewählten ZP_KS [mm ×100]	
	[ZP_Y]	Y – Koordinate der Zielposition im ausgewählten ZP_KS [mm ×100]	
	[ZP_Z]	Z – Koordinate der Zielposition im ausgewählten ZP_KS [mm ×100]	
	[Status]	0	kein Fehler
		1	fehlerhafte Auswahl mindestens eines Parameters

Tabelle 21: Syntax der Botschaft *Position anfahren*

Der Parameter *[OffsetZ]* bewirkt eine Verschiebung der Zielposition in Stoßrichtung des Roboterwerkzeugs. Da die Messung bei unbelastetem Kraftsensor gestartet werden muss, ist dies besonders bei anschließenden Steifigkeitsmessungen nützlich.

### Variante 1: Auf Basis ausgelesener Positionsdaten

Über die Botschaft *Position auslesen* empfangene Koordinaten und Rotationswinkel sind direkt als Eingangsparameter für die Botschaft *Position anfahren* verwendbar. Auf eine übereinstimmende Auswahl des Koordinatensystems ist zu achten.

### Variante 2: Auf Basis der Punkt – Richtungsform

Liegt die Zielposition des Roboterwerkzeugs in Form eines Punkt – Richtungsvektors vor, wie es beim Datenexport aus CAD – Anwendungen der Fall ist, muss der Richtungsvektor *D* aus dem kartesischen Koordinatensystem ins Kugelkoordinatensystem umgerechnet werden. Die Koordinaten X, Y und Z des Zielpunkts bleiben unverändert.

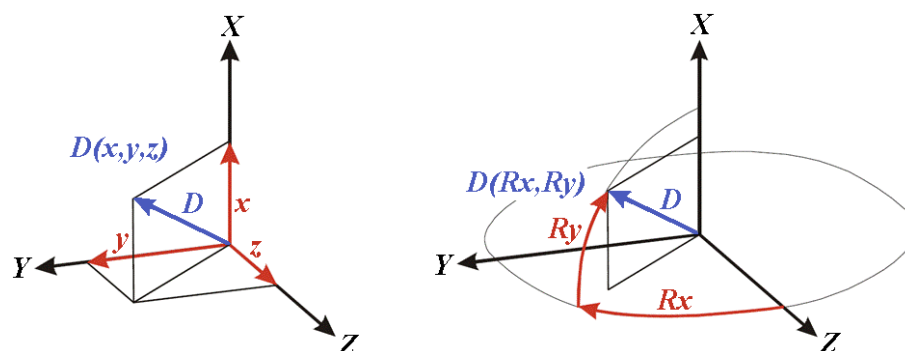


Abbildung 67: Gegenüberstellung von kartesischen Koordinaten und Kugelkoordinaten

Die Rotationswinkel *Rx* und *Ry* lassen sich anhand der folgenden Formeln bestimmen.

$$R_x = \arctan\left(\frac{y}{z}\right)$$

Formel 9: Rotationswinkel *Rx* aus kartesischen Koordinaten

$$R_y = \arctan\left(\frac{x}{\sqrt{y^2 + z^2}}\right)$$

**Formel 10: Rotationswinkel  $R_y$  aus kartesischen Koordinaten**

Die Drehung  $R_z$  ist bei rotationssymmetrischen Prüfstempeln nicht relevant.

#### 4.6.3.2.2 Achsenbewegungen

##### Achsenbewegung

Durch die Botschaft *Achsenbewegung* können die Roboterachsen einzeln oder gemeinsam bewegt werden.

S)	<b>Syntax: Y[J1];[J2];[J3];[J4];[J5];[J6];[ZP_KS];[Geschw];[Beschl];[Bremsen]!</b>	
	[J1]	Winkeländerung der Achse 1 [° × 100]
	⋮	⋮
	[J6]	Winkeländerung der Achse 6 [° × 100]
	[ZP_KS]	0 Ausgabe der Zielposition im Roboterkoordinatensystem
		1 Ausgabe der Zielposition im Bauteilkoordinatensystem
		2 Ausgabe der Zielposition im Türkoordinatensystem
		3 Ausgabe der Zielposition im Fahrzeugkoordinatensystem
	[Geschw]	Geschwindigkeit der Roboterbewegung [%]
	[Beschl]	positive Beschleunigung der Roboterbewegung [%]
	[Bremsen]	negative Beschleunigung der Roboterbewegung [%]
E)	<b>Syntax: y[ZP_X];[ZP_Y];[ZP_Z];[Status]&lt;CR&gt;</b>	
	[ZP_X]	X – Koordinate der Zielposition im ausgewählten ZP_KS [mm × 100]
	[ZP_Y]	Y – Koordinate der Zielposition im ausgewählten ZP_KS [mm × 100]
	[ZP_Z]	Z – Koordinate der Zielposition im ausgewählten ZP_KS [mm × 100]
	[Status]	0 kein Fehler
		1 fehlerhafte Auswahl des Parameters [ZP_KS]

**Tabelle 22: Syntax der Botschaft *Achsenbewegung***

Die Zahlenwerte der Parameter [J1] bis [J6] bestimmen die Winkeländerung der jeweiligen Roboterachsen. Die Drehrichtung ist durch das Vorzeichen festgelegt.

##### Werkzeug ausrichten

Diese Botschaft bewirkt eine Ausrichtung des Roboterwerkzeugs um seine Stoßrichtung. Die Anwendung beschränkt sich im Wesentlichen darauf, das Roboterwerkzeug nach einer Positionierung über die Botschaft *Position anfahren* in eine definierte Ausrichtung zu bewegen.

S)	<b>Syntax: L[Ausrichtung];[ZP_KS];[Geschw];[Beschl];[Bremsen]!</b>	
	[Ausrichtung]	Ausrichtung des Werkzeugs in der X – Y – Ebene des Werkzeug – KS [ $^{\circ} \times 10$ ]
	[ZP_KS]	0 Ausgabe der Zielposition im Roboterkoordinatensystem
		1 Ausgabe der Zielposition im Bauteilkoordinatensystem
		2 Ausgabe der Zielposition im Türkoordinatensystem
		3 Ausgabe der Zielposition im Fahrzeugkoordinatensystem
	[Geschw]	Geschwindigkeit der Roboterbewegung [%]
	[Beschl]	positive Beschleunigung der Roboterbewegung [%]
	[Bremsen]	negative Beschleunigung der Roboterbewegung [%]
E)	<b>Syntax: I[ZP_X];[ZP_Y];[ZP_Z];[Status]&lt;CR&gt;</b>	
	[ZP_X]	X – Koordinate der Zielposition im ausgewählten ZP_KS [mm $\times 100$ ]
	[ZP_Y]	Y – Koordinate der Zielposition im ausgewählten ZP_KS [mm $\times 100$ ]
	[ZP_Z]	Z – Koordinate der Zielposition im ausgewählten ZP_KS [mm $\times 100$ ]
	[Status]	0 kein Fehler
		1 fehlerhafte Auswahl des Parameters [ZP_KS]

Tabelle 23: Syntax der Botschaft *Werkzeug ausrichten*

#### 4.6.3.2.3 Linearbewegungen

##### Linearbewegung

Diese Botschaft generiert eine lineare Bewegung des Roboterwerkzeugs.

S)	<b>Syntax: Z[StreckeXY];[RichtungXY];[StreckeZ];[RotationX];[RotationY];[RotationZ];[ZP_KS];[Geschw];[Beschl];[Bremsen]!</b>	
	[StreckeXY]	Strecke in der X – Y – Ebene des Werkzeug – KS [mm $\times 100$ ]
	[RichtungXY]	Bewegungsrichtung in der X – Y – Ebene des Werkzeug – KS [ $^{\circ} \times 10$ ]
	[StreckeZ]	Strecke in Stoßrichtung des Werkzeugs [mm $\times 100$ ]
	[RotationX]	Rotation um senkrechte Achse [ $^{\circ} \times 10$ ]
	[RotationY]	Rotation um waagerechte Achse [ $^{\circ} \times 10$ ]
	[RotationZ]	Rotation um Stoßrichtung des Werkzeugs [ $^{\circ} \times 10$ ]
	[ZP_KS]	0 Ausgabe der Zielposition im Roboterkoordinatensystem
		1 Ausgabe der Zielposition im Bauteilkoordinatensystem
		2 Ausgabe der Zielposition im Türkoordinatensystem
		3 Ausgabe der Zielposition im Fahrzeugkoordinatensystem
	[Geschw]	Geschwindigkeit der Roboterbewegung [%]
	[Beschl]	positive Beschleunigung der Roboterbewegung [%]
	[Bremsen]	negative Beschleunigung der Roboterbewegung [%]
E)	<b>Syntax: z[ZP_X];[ZP_Y];[ZP_Z];[Status]&lt;CR&gt;</b>	
	[ZP_X]	X – Koordinate der Zielposition im ausgewählten ZP_KS [mm $\times 100$ ]
	[ZP_Y]	Y – Koordinate der Zielposition im ausgewählten ZP_KS [mm $\times 100$ ]
	[ZP_Z]	Z – Koordinate der Zielposition im ausgewählten ZP_KS [mm $\times 100$ ]
	[Status]	0 kein Fehler
		1 fehlerhafte Auswahl des Parameters [ZP_KS]

Tabelle 24: Syntax der Botschaft *Linearbewegung*

Dabei wird die Bewegungsrichtung durch Parameter definiert, die sich aus der Perspektive des Bedieners auf das Roboterwerkzeug ableiten. Abbildung 68 zeigt die Lage der zur Berechnung der Zielposition relevanten Parameter.

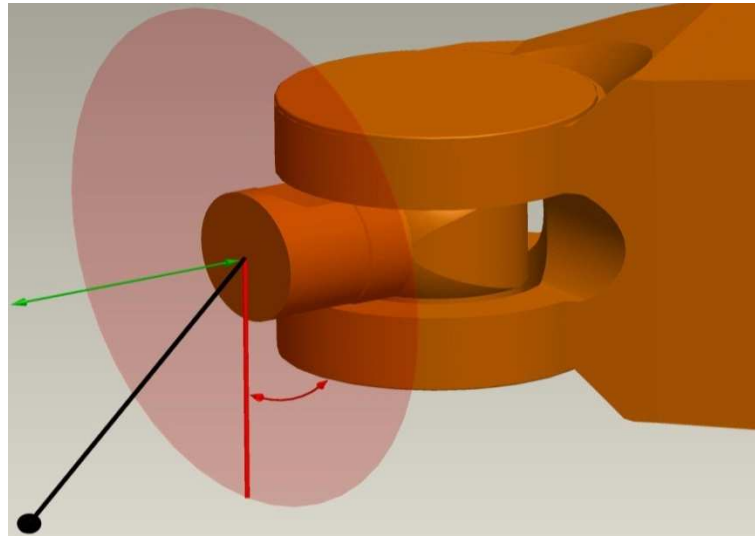


Abbildung 68: Darstellung der Parameter zur Botschaft *Linearbewegung*

Durch den Parameter *[StreckeXY]* wird der rot markierte Radius des hellroten Kreises in der X – Y – Ebene des Werkzeugkoordinatensystems festgelegt. Der rot markierte Pfeil in dieser Ebene steht für den Parameter *[RichtungXY]* und legt die Bewegungsrichtung in der X – Y – Ebene fest. Die in Abbildung 68 dargestellte Ausrichtung des Radius entspricht einem Winkelwert von 180 Grad für *[RichtungXY]*. Zusammen mit dem grün dargestellten Parameter *[StreckeZ]* lässt sich die schwarz markierte Bahnkurve zur Zielposition generieren.

### Kraftbegrenzte Stoßbewegung

Die Botschaft *kraftbegrenzte Stoßbewegung* erzeugt eine Bewegung in Stoßrichtung des Werkzeugs bis zum Erreichen der angegebenen Maximalkraft.

S)	<b>Syntax: O[StreckeZ];[KraftWert];[ZP_KS];[Geschw];[Beschl];[Bremsen]!</b>	
	[StreckeZ]	Strecke in Stoßrichtung des Werkzeugs [mm ×100]
	[KraftWert]	Maximalkraft [N ×10]
	[ZP_KS]	0 Ausgabe der Zielposition im Roboterkoordinatensystem
		1 Ausgabe der Zielposition im Bauteilkoordinatensystem
		2 Ausgabe der Zielposition im Türkoordinatensystem
		3 Ausgabe der Zielposition im Fahrzeugkoordinatensystem
	[Geschw]	Geschwindigkeit der Roboterbewegung [mm/s ×10]
	[Beschl]	positive Beschleunigung der Roboterbewegung [%]
	[Bremsen]	negative Beschleunigung der Roboterbewegung [%]
E)	<b>Syntax: o[ZP_X];[ZP_Y];[ZP_Z];[Status]&lt;CR&gt;</b>	
	[ZP_X]	X – Koordinate der Zielposition im ausgewählten ZP_KS [mm ×100]
	[ZP_Y]	Y – Koordinate der Zielposition im ausgewählten ZP_KS [mm ×100]
	[ZP_Z]	Z – Koordinate der Zielposition im ausgewählten ZP_KS [mm ×100]
	[Status]	0 kein Fehler
		1 fehlerhafte Auswahl des Parameters <i>[ZP_KS]</i>

Tabelle 25: Syntax der Botschaft *Kraftbegrenzte Stoßbewegung*

Der Einsatz ist für Steifigkeitsprüfungen und kraftgeregelte Dauerbetätigungen konzipiert. Nach Erreichen der Maximalkraft bewegt sich der Roboterarm automatisch zur Ausgangsposition zurück. Wird die Maximalkraft nicht erreicht, ist die Länge der Stoßbewegung durch den Parameter *[StreckeZ]* begrenzt. Die Bewegungsrichtung ist vom Vorzeichen des Parameters *[StreckeZ]* abhängig, ein positives Vorzeichen steht für eine Druckbewegung. Vor dem Start des Bewegungsablaufs wird der Messdatenexport automatisch aktiviert und beim Erreichen der Ausgangsposition wieder deaktiviert.

#### 4.6.3.2.4 Kreisbewegungen

Kreisförmige Bewegungen von Knickarmrobotern sind in der herstellerseitigen Programmiersprache auf dem Robotercontroller durch drei Positionen im Raum definiert. Über die Parameter der nachfolgenden Botschaften können diese Positionen unter Berücksichtigung des ausgewählten Koordinatensystems berechnet werden.

##### Variante 1: Kreisbewegung

Diese Botschaft erzeugt eine kreisförmige Bewegung des Werkzeugs im Raum. Dabei wird die Kreisbahn durch Parameter definiert, deren Inhalt sich aus der Perspektive des Bedieners auf das Roboterwerkzeug ergibt.

S)	<b>Syntax: R[Radius];[Winkel];[MittelPLage];[MittelPVers];[KraftB];[KraftWert];[ZP_KS];[Geschw];[Beschl];[Bremsen]!</b>	
	[Radius]	Radius der Kreisbewegung [mm × 100]
	[Winkel]	Mittelpunktswinkel der Kreisbewegung [° × 100]
	[MittelPLage]	Lage des Kreismittelpunkts in der X – Y – Ebene des Werkzeug – KS [° × 10]
	[MittelPVers]	Verschiebung des Kreismittelpunkts in Stoßrichtung des Werkzeugs [mm × 100]
	[KraftB]	0 Kraftbegrenzung deaktivieren
		1 Kraftbegrenzung aktivieren
	[KraftWert]	Maximalkraft [N × 10]
	[ZP_KS]	0 Ausgabe der Zielposition im Roboterkoordinatensystem
		1 Ausgabe der Zielposition im Bauteilkoordinatensystem
		2 Ausgabe der Zielposition im Türkoordinatensystem
		3 Ausgabe der Zielposition im Fahrzeugkoordinatensystem
	[Geschw]	Geschwindigkeit der Roboterbewegung [mm/s × 10]
	[Beschl]	positive Beschleunigung der Roboterbewegung [%]
	[Bremsen]	negative Beschleunigung der Roboterbewegung [%]
E)	<b>Syntax: r[ZP_X];[ZP_Y];[ZP_Z];[Status]&lt;CR&gt;</b>	
	[ZP_X]	X – Koordinate der Zielposition im ausgewählten KS [mm × 100]
	[ZP_Y]	Y – Koordinate der Zielposition im ausgewählten KS [mm × 100]
	[ZP_Z]	Z – Koordinate der Zielposition im ausgewählten KS [mm × 100]
	[Status]	0 kein Fehler
		1 fehlerhafte Auswahl mindestens eines Parameters

Tabelle 26: Syntax der Botschaft *Parameterbasierte Kreisbewegung*



Abbildung 69 beschreibt die zur Berechnung der Bahnkurve relevanten Parameter.

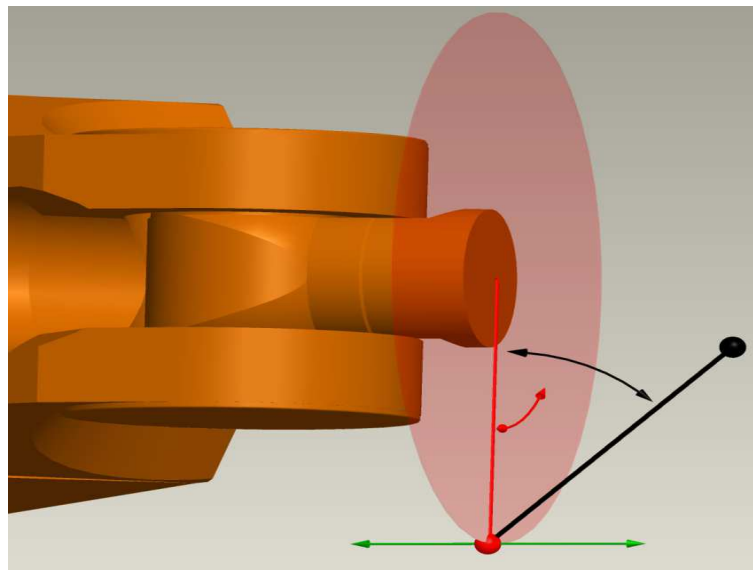


Abbildung 69: Darstellung der Parameter zur Botschaft *Kreisbewegung*

Mit Hilfe der rot markierten Parameter *[Radius]* und *[MittelPLage]* ist der Mittelpunkt der Kreisbewegung in der hellrot eingefärbten X – Y – Ebene des Werkzeugkoordinatensystems definiert. Der dargestellte Winkel für *[MittelPLage]* in der Abbildung 69 entspricht einem Wert von 180 Grad. Eine Verschiebung des Kreismittelpunkts in Z – Richtung des Werkzeugkoordinatensystems ist über den grün gekennzeichneten Parameter *[MittelPVers]* möglich. Über den schwarz eingefärbten Mittelpunktswinkel, übertragen durch den Parameter *[Winkel]*, ist die Bogenlänge zur Zielposition definiert. Das Vorzeichen von *[Winkel]* entscheidet über eine Zug- oder Druckbewegung.

Durch eine Aktivierung der Kraftbegrenzung mit Hilfe des Parameters *[KraftB]* lassen sich analog zur Botschaft *kraftbegrenzte Stoßbewegung* Steifigkeitsuntersuchungen oder kraftge-regelte Dauerversuche durchführen. Das Roboterwerkzeug wird nach dem Erreichen der Maximalkraft im Kreisbogen zur Ausgangsposition zurückbewegt. Ohne Kraftbegrenzung verbleibt das Werkzeug an der Zielposition. Wird die Maximalkraft nicht erreicht, ist die Strecke der Bahnkurve durch die Bogenlänge begrenzt.

### Variante 2: Drehung im Türkoordinatensystem

Diese Botschaft bewirkt eine Drehung der Werkzeugposition um die X – Achse des Türkoordinatensystems.

S>	<b>Syntax: V[Winkel];[ZP_KS];[Geschw];[Beschl];[Bremsen]!</b>	
	[Winkel]	Mittelpunktswinkel um die X – Achse des Tür – KS [ $^{\circ} \times 100$ ]
	[ZP_KS]	0 Ausgabe der Zielposition im Roboterkoordinatensystem
		1 Ausgabe der Zielposition im Bauteilkoordinatensystem
		2 Ausgabe der Zielposition im Türkoordinatensystem
		3 Ausgabe der Zielposition im Fahrzeugkoordinatensystem
	[Geschw]	Geschwindigkeit der Roboterbewegung [%]
	[Beschl]	positive Beschleunigung der Roboterbewegung [%]
	[Bremsen]	negative Beschleunigung der Roboterbewegung [%]


	<b>Syntax:</b> v[ <i>ZP_X</i> ];[ <i>ZP_Y</i> ];[ <i>ZP_Z</i> ];[ <i>Status</i> ]<CR>		
	[ <i>ZP_X</i> ]	X – Koordinate der Zielposition im ausgewählten KS [mm ×100]	
	[ <i>ZP_Y</i> ]	Y – Koordinate der Zielposition im ausgewählten KS [mm ×100]	
	[ <i>ZP_Z</i> ]	Z – Koordinate der Zielposition im ausgewählten KS [mm ×100]	
	[ <i>Status</i> ]	0	kein Fehler
		1	Fehlerhafte Auswahl des Parameters [ <i>ZP_KS</i> ]

Tabelle 27: Syntax der Botschaft *Drehung im Türkoordinatensystem*

Mittels dieser Botschaft ist eine zeitoptimierte Erstellung äußerst präziser Bewegungen um eine feste Drehachse mit einstellbarem Bewegungswinkel möglich. Als Drehachse fungiert dabei die X – Achse des Türkoordinatensystems. Das Prüfmodul *Türbewegung*, welches in Abschnitt 5.2.1 detailliert beschrieben wird, beinhaltet mit den Roboterschritten *Tür in Zwischenposition*, *Tür aufschwenken* und *Tür schließen* drei Drehungen um die Scharnierachse. Bei den genannten Bewegungsabläufen lassen sich neun ansonsten mit dem Teach – In – Verfahren einzulernende Positionen durch drei Winkelparameter ersetzen. Die Drehrichtung ist über das Vorzeichen des Parameters [*Winkel*] definiert.

#### 4.6.4 Umsetzung auf dem Robotercontroller

Das Kommunikationsprotokoll definiert, welche Parameter zur Ausführung von Betätigungen bei zeitgleicher Messdatenerfassung vom Prüfstandsrechner zum Robotercontroller übertragen werden müssen. Der Programmcode zur Umsetzung dieser Parameter in eine Roboterbewegung sowie die Kommunikation mit externer Messhardware muss über die herstellerspezifische Programmierungsumgebung erfolgen. Im Einzelfall entscheiden der verfügbare Funktionsumfang und die Leistungsfähigkeit des Robotercontrollers über eine vollständige oder nur teilweise Anwendbarkeit des Kommunikationsprotokolls.

Nach dem Start des Programms auf dem Robotercontroller, welcher zur Vereinfachung der Bedienung idealerweise automatisch nach dem Einschalten erfolgt, findet eine Initialisierung statt. Hierbei werden sämtliche Einstellungen auf Standardwerte gesetzt, möglicherweise noch gespeicherte Bewegungsabläufe gelöscht und der vorliegende Betriebszustand an den Prüfstandsrechner transferiert. Nach Beendigung der Initialisierung starten sechs Tasks mit aufgabenabhängiger Priorität. Der Task *Lesen* überwacht den Empfang von Botschaften des Kommunikationsprotokolls und startet, entsprechend dem charakteristischen ersten Buchstaben, das jeweilige Unterprogramm. Eine hier erzeugte Antwortbotschaft sowie die übrigen zu versendenden Daten werden in einer Queue gesammelt und vom Task *Schreiben* zum Prüfstandsrechner übertragen. Die Tasks *Status* und *Fehler* überwachen permanent die Zustände des Controllers und generieren ereignisgesteuert Meldungen an den Prüfstandsrechner oder auf dem Display des Handbediengeräts. Bis der Messdaten- oder Achsmomentexport durch entsprechende Botschaften aktiviert wird, pausieren die Tasks *Messdaten* und *Achslast*.

Die Messdatenerfassung samt Übertragung an den Prüfstandsrechner ist zeitkritisch, daher besitzen die involvierten Tasks *Schreiben* und *Messdaten* die höchste Priorität 1. Da die Priorität des Tasks *Lesen* einen Einfluss auf die Verarbeitungsgeschwindigkeit der Unterprogramme hat, besitzt sie den Wert 2. Die übrigen Tasks sind nicht zeitkritisch und daher untergeordnet. Abbildung 70 zeigt die Softwarestruktur sowie den durch Pfeile dargestellten Informationsfluss zwischen den Unterprogrammen und den Tasks auf dem Robotercontroller. Blaue Pfeile symbolisieren den einmaligen Datenaustausch beim Empfangen einer Botschaft, grüne Pfeile stehen für eine kontinuierliche Datenübertragung im Bedarfsfall. Rot signalisiert eine ereignisgesteuerte Versendung von Informationen.

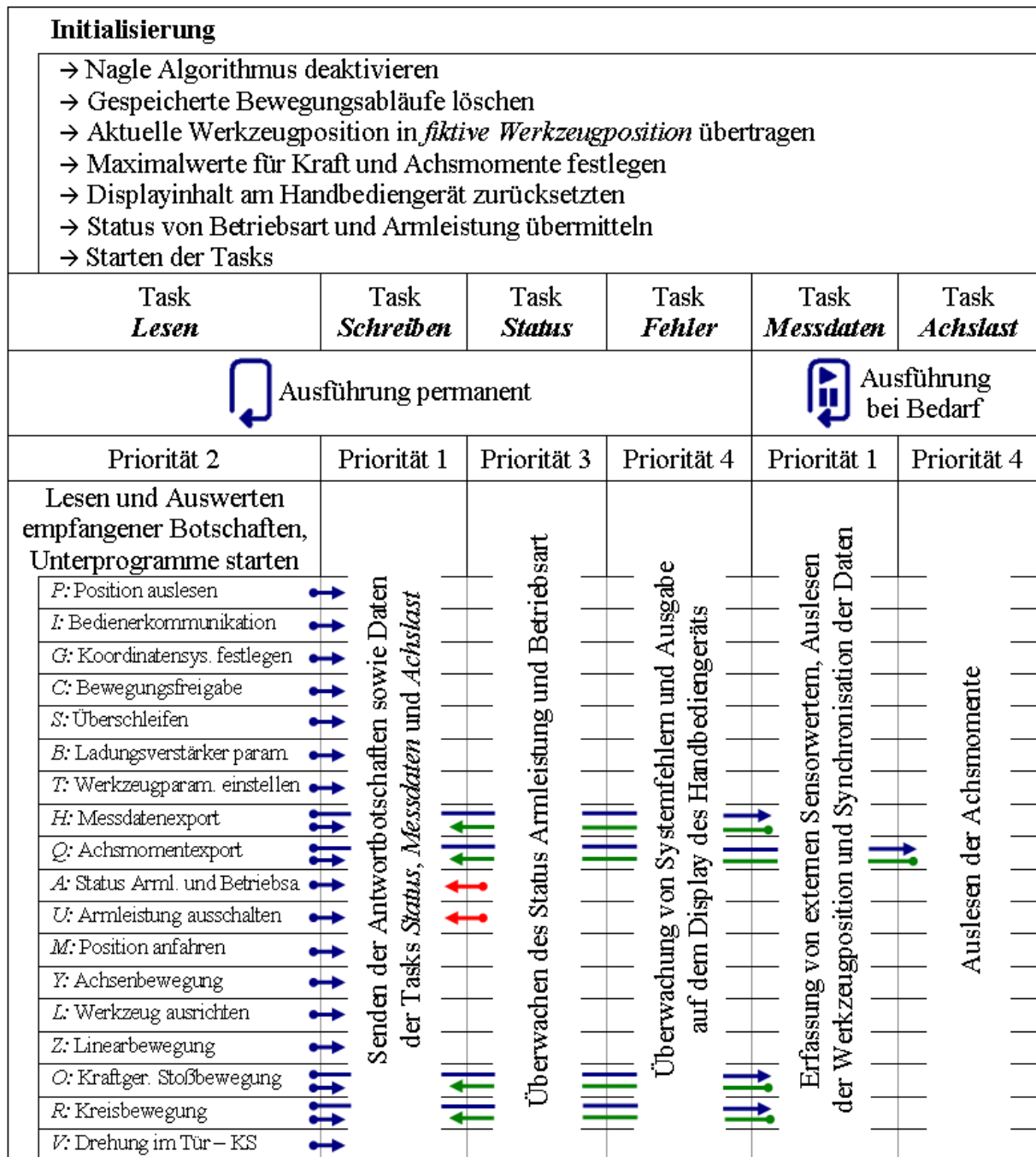


Abbildung 70: Tasks auf dem Robotercontroller

Bei den Bewegungsbotschaften *Werkzeug ausrichten*, *Linearbewegungen* und *Kreisbewegungen* erfolgt die Berechnung der Bahnkurve im Werkzeugkoordinatensystem auf Basis der *fiktiven Werkzeugposition*. Die mit den Botschaften übertragenen Winkelparameter [*Ausrichtung*], [*RichtungXY*] und [*MittelPLage*] beinhalten Werte, die der Bediener bei einer Betrachtung des Roboterwerkzeugs ohne Kenntnis und Berücksichtigung der Lage von Koordinatensystemen festlegt. Die Winkelwerte haben aus der Sicht des Bedieners primär einen Bezug zur Positionierung des Roboterwerkzeugs gegenüber dem zu bewegenden Bauteil. Das folgende Beispiel soll diesen Zusammenhang verdeutlichen.

Der Wert 0 Grad des Parameters [*RichtungXY*] der Botschaft *Linearbewegung* soll bewirken, dass sich das Werkzeug in der X – Y – Ebene des Werkzeugkoordinatensystems um den Wert des Parameters [*StreckeXY*] nach *oben* bewegt. Für *oben* ist in diesem Fall die Bewegungsrichtung gesucht, welche in der X – Y – Ebene des Werkzeugkoordinatensystems zur höchst-

möglichen Z – Koordinate im Roboterkoordinatensystem führt. Umgekehrt resultiert aus dem Wert 180 Grad eine Bewegung in der X – Y – Ebene des Werkzeugkoordinatensystems zur kleinstmöglichen Z – Koordinate im Roboterkoordinatensystem. Je kleiner der Schnittwinkel zwischen der X – Y – Ebene des Werkzeugkoordinatensystems und der Z – Achse des Roboterkoordinatensystems ausfällt, desto senkrechter nach oben verläuft die Bewegung bei einem Winkel von 0 Grad für den Parameter *[RichtungXY]*.

Zur Berechnung einer Bahnkurve auf Basis der beschriebenen Parameter ist ein Zusammenhang zwischen der für den Bediener nicht sichtbaren Verdrehung des Werkzeugkoordinatensystems um die Stoßrichtung sowie der Sichtweise auf das reale Roboterwerkzeug herzustellen. Dazu ist es zielführend, am Werkzeug des Roboters ein Hilfskoordinatensystem zu installieren, bei dem die X – Achse einen möglichst minimalen Schnittwinkel zur Z – Richtung des Roboterkoordinatensystems aufweist. Für diesen Zweck wird ein definierter Punkt auf der X – Achse des Werkzeugkoordinatensystems in 3.600 Schritten um die Stoßrichtung rotiert. Mit jeder Iteration erfolgt eine Transformation des neu erhaltenen Punkts ins Roboterkoordinatensystem samt Vergleich der dortigen Z – Koordinate mit dem bisherigen Maximum. Aus der Nummer des Iterationsschritts mit der maximalen Z – Koordinate im Roboterkoordinatensystem geht der gesuchte Verdrehwinkel hervor, der benötigt wird, um auf Basis des Werkzeugkoordinatensystems das Hilfskoordinatensystem einzustellen. Im Anschluss ist die Generierung der Bahnkurve mit Hilfe der Parameter möglich.

#### 4.6.5 Inbetriebnahme einfachster Betätigungen

Die Bedienoberfläche zur Einstellung *einfachster Betätigungen* ist so konzipiert, dass Versuchingenieure mechanische Dauerversuche oder Einzelmessungen mit sehr begrenztem Bewegungsumfang ohne Hilfe von Softwareexperten einstellen können. Dafür sind keine umfangreichen Kenntnisse über Koordinatensysteme, sequenzielle Abfolgen der Programmierung oder Eigenschaften von Roboterwerkzeugen erforderlich. Lediglich das manuelle Positionieren des Roboterarms mittels eines Handbediengeräts wird als Qualifikation vorausgesetzt. Diese Fähigkeit ist mit überschaubarem Aufwand von weniger als einer Stunde erlernbar.

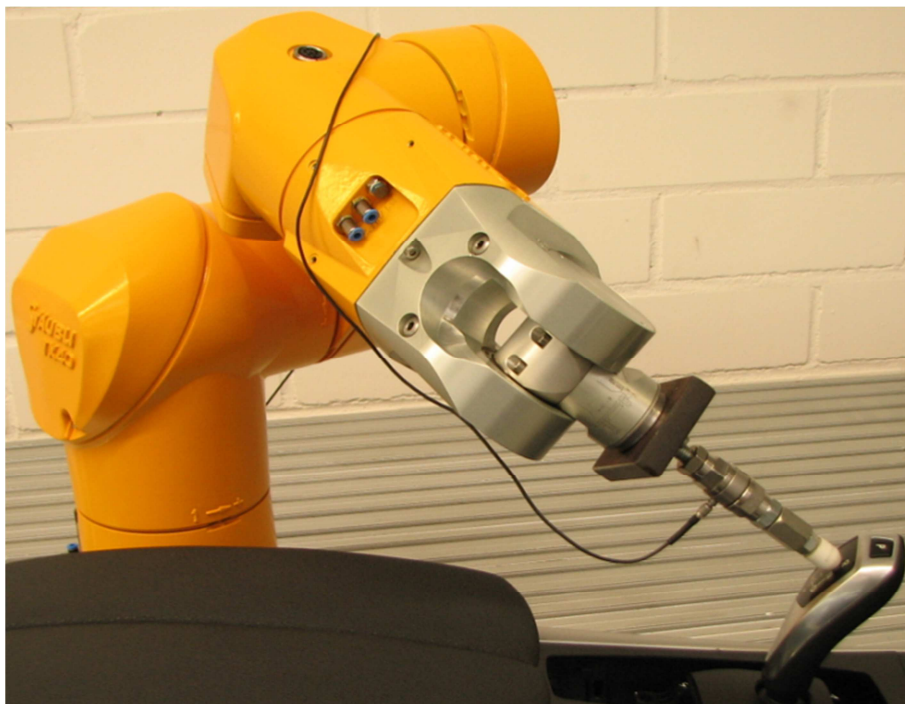


Abbildung 71: Mechanischer Aufbau zur Betätigung eines Gangwahlhebels

Als konkretes Beispiel für eine einfachste Betätigung dient die zyklische Belastung eines Gangwahlhebels durch das Aufbringen einer definierten Kraft. Der mechanische Aufbau ist in Abbildung 71 dargestellt.

Zur Einstellung eines Bewegungsablaufs stehen bei einfachsten Betätigungen sechs sogenannte *Prüfmodule* zur Verfügung. Die Module *Punkt anfahren*, *Linearbewegung* und *Kreisbewegung* gründen auf den gleichnamigen Botschaften des Kommunikationsprotokolls. Das Modul *Punkt anfahren* wird daher nur benötigt, sobald verschiedene Ausgangspositionen für die eigentlichen Prüfbetätigungen erforderlich sind, beispielsweise um an einem Lenkrad mehrere nebeneinander positionierte Knöpfe im Rahmen eines Dauerlaufs zu prüfen. Die Prüfbetätigungen werden je nach Bewegungscharakteristik des Prüflings als kraftgeregelte oder wegbegrenzte Linear- oder Kreisbewegung ausgeführt. Die Prüfmodule *Eingang lesen* oder *Ausgang setzen* kommen zur Anwendung, um das Erreichen von Endstellungen über Näherungssensoren abzufragen oder um zusätzliche Aktoren anzusteuern.



Abbildung 72: Prüfmodule zur Ausführung einfachster Bewegungen

Vor der Übernahme eines Prüfmoduls in die *Dauerlaufkonfiguration* erscheint ein Fenster zur Einstellung der modulabhängigen Parameter, mit deren Hilfe auf Basis der aktuellen Position des Roboterwerkzeugs die Bahnkurve der Betätigung berechnet wird. Abbildung 73 zeigt die Bedienoberfläche zur Parametrierung des Prüfmoduls *Kreisbewegung*.



Abbildung 73: Bedienoberfläche zur Parametrierung einer Kreisbewegung

Die Einstellung der Parameter wird durch diese Maske vereinfacht. Alternativ zum Mittelpunktswinkel kann die maximale Bogenlänge angegeben werden. Das korrekte Vorzeichen für den Parameter *[Winkel]* der Botschaft *Kreisbewegung* wird über das Eingabefeld *Richtung*



festgelegt. Der Parameter *[MittelPLage]* wird nicht als Zahlenwert erwartet, die Einstellung erfolgt über einen Drehknopf. Diese grafische Definition erleichtert die Nachvollziehbarkeit für den Bediener. Bei aktivierten Messungen wird automatisch ein Kraft – Weg – Diagramm aufgenommen und darauf basierend die Betätigungsenergie berechnet. In definierbaren Grenzen ist die Durchführung von Steifigkeitsberechnungen optional möglich. Nach dem Einstellen der Parameter wird das Prüfmodul in die Dauerlaufkonfiguration übernommen.

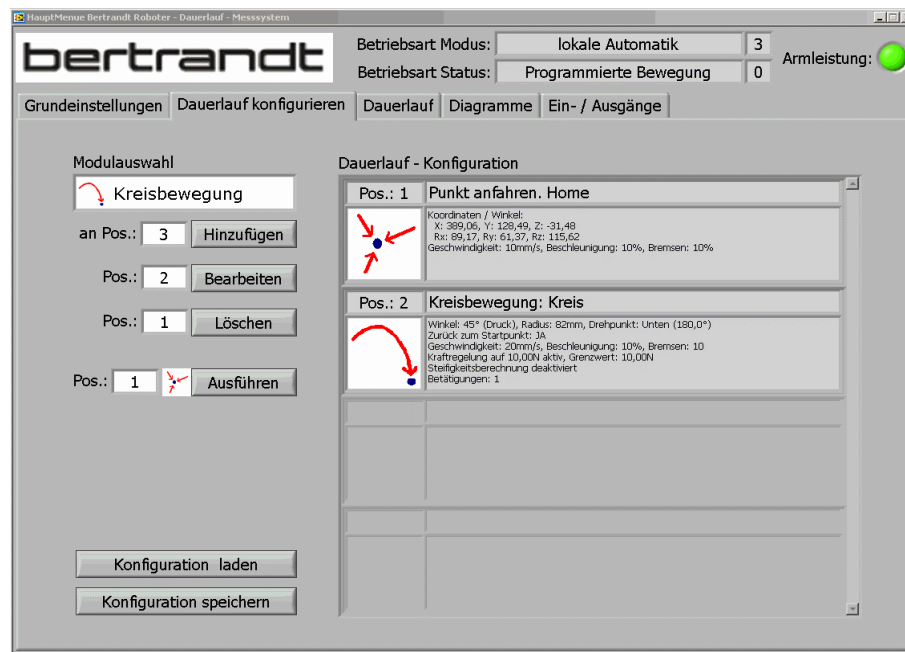


Abbildung 74: Bedienoberfläche zur Konfiguration einfacher Bewegungen

Über die in Abbildung 74 dargestellte Bedienoberfläche lassen sich die in Abbildung 72 aufgeführten *Prüfmodule* in wenigen Minuten zu einem Dauerlauf zusammenstellen. Im Anschluss kann die zyklische Erprobung über den Reiter *Dauerlauf* gestartet, beendet oder zeitweilig unterbrochen werden. Die Bedienoberfläche weist deutliche Parallelen zur Prüfablaufsteuerung auf. Dadurch erhöhen sich der Wiedererkennungswert und die Sicherheit der Bediener in der praktischen Anwendung.

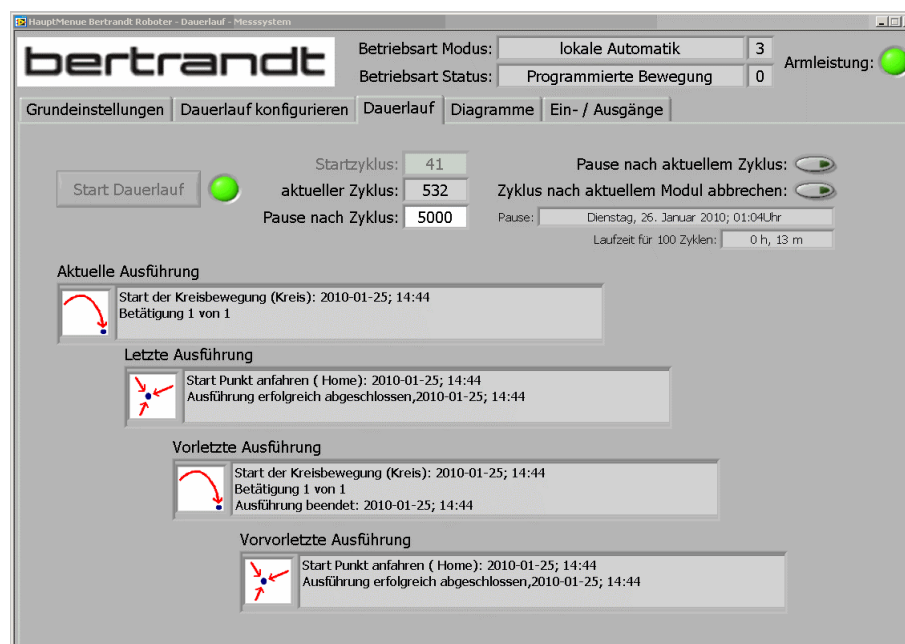


Abbildung 75: Bedienoberfläche zur Steuerung einfacher Bewegungsabläufe



Im Reiter *Diagramme* stehen die Messkurven zur Verfügung, die üblicherweise in der Komponentenerprobung der Bewertung und Versuchsdokumentation dienen. Abbildung 76 zeigt den charakteristischen Wert der Betätigungsenergie, welche zur Belastung des Prüflings bis zur eingestellten Kraft im jeweiligen Betätigungszyklus notwendig ist.

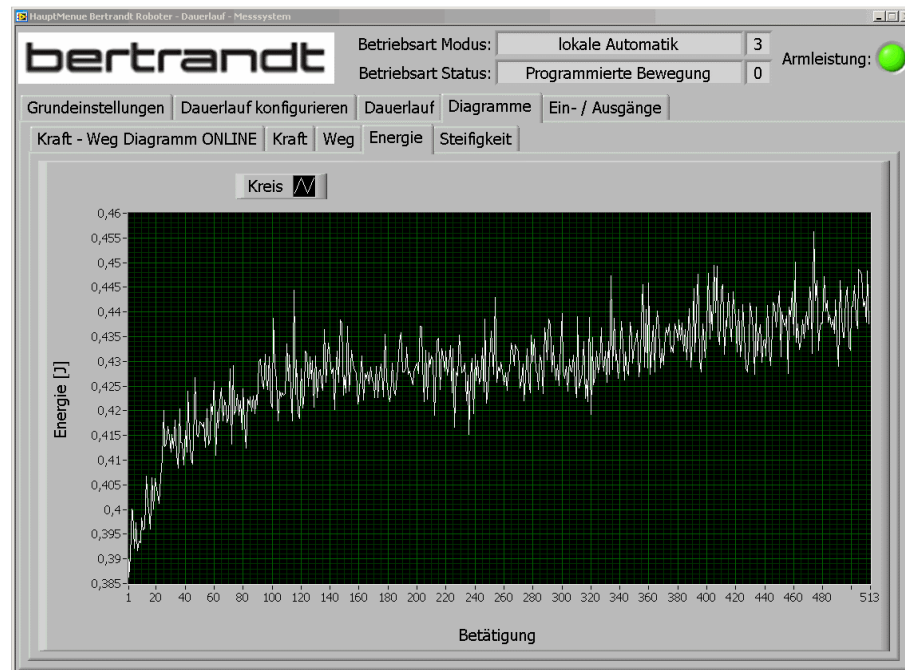
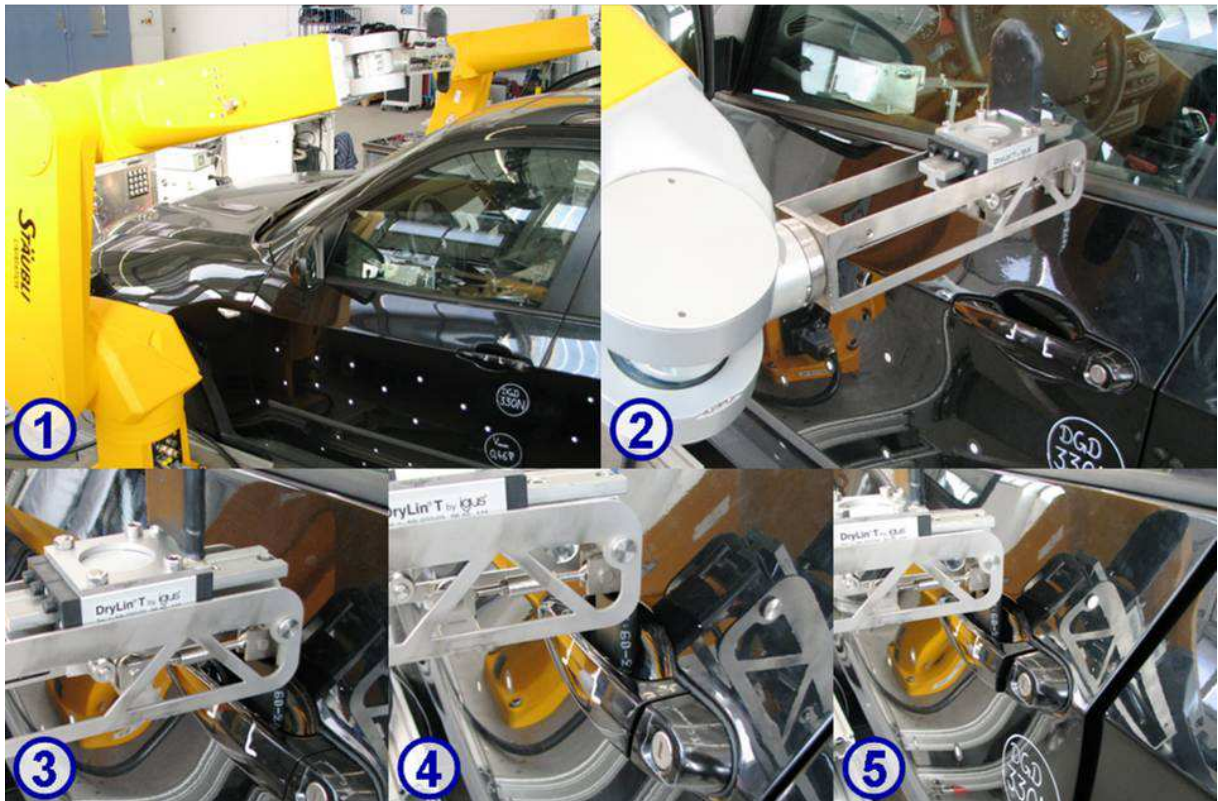


Abbildung 76: Bedienoberfläche der Messdatenanzeige

Die Bedienoberfläche zur Inbetriebnahme einfachster Betätigungen ist in die Gesamtheit des Prüfsystemkonzepts als Softwaretool integriert und über diese kann auf die Inhalte der Datenbasis zugegriffen werden. Die Konzeption ist unabhängig vom Roboterhersteller, sofern das Kommunikationsprotokoll der aufgabenorientierten Schnittstelle anwendbar ist. Eine Beschränkung auf wenige im Versuchsumfeld typische Funktionen sorgt für eine sehr übersichtlich gestaltete Bedienoberfläche. Der Schulungsaufwand für unerfahrene Mitarbeiter beträgt nur etwa drei Stunden.

#### 4.6.6 Parametrierung komplexer Bewegungsabläufe

*Komplexe Bewegungsabläufe* im Kontext der aufgabenorientierten Schnittstelle setzen sich aus Botschaften des Kommunikationsprotokolls und mathematischen Operationen der Vektorrechnung zusammen. Die Abfolge ist in einem Grundgerüst definiert, welches für eine konkrete Prüfaufgabe durch Softwareexperten zusammengestellt wird. Der Bewegungsablauf basiert grundsätzlich auf einem zweckmäßig gewählten Koordinatensystem. Zur Verbesserung der Verständlichkeit für den Bediener werden diese komplexen Bewegungsabläufe als *Roboterschritte* abstrahiert. Abbildung 77 zeigt beispielhaft den Bewegungsablauf des Roboterschritts *Tür entriegeln Roboter außen*.

Abbildung 77: Bewegungsablauf des Roboterschritts *Tür entriegeln Roboter außen*

Hinter diesem Bewegungsablauf verbirgt sich das in Tabelle 28 dargestellte Grundgerüst.

Botschaft / Operation	Beschreibung und Parametrierung
<i>Bewegungsfreigabe</i>	Bewegungsfreigabe sofort entziehen
<i>Überschleifen</i>	Überschleifen aktivieren (5mm)
<i>Position anfahren</i> ①	Bewegung zur Ausgangsposition
<i>Position anfahren</i> ②	Bewegung zur Position oberhalb des Türgriffs
<i>Linearbewegung</i> ③	Bewegung zur Einfahrt in den Türgriff
<i>Bewegungsfreigabe</i>	Bewegungsfreigabe erteilen und nach Bewegungsende entziehen
Roboter führt Bewegungsablauf ① bis ③ aus	
<i>Überschleifen</i>	Überschleifen aktivieren (2mm)
<i>Linearbewegung</i> ④	Bewegung an den Türgriff
<i>Kreisbewegung</i> ⑤	Bewegung zum Entriegeln der Tür
<i>Messdatenexport</i>	Messdatenexport aktivieren
<i>Bewegungsfreigabe</i>	Bewegungsfreigabe erteilen und nach Bewegungsende entziehen
Roboter führt Bewegungsablauf ④ und ⑤ aus	
<i>Messdatenexport</i>	Messdatenexport deaktivieren
<i>Überschleifen</i>	Überschleifen deaktivieren
<i>Rechenoperation</i>	Berechnung des Türöffnungswinkels

Tabelle 28: Grundgerüst des Roboterschritts *Tür entriegeln Roboter außen*

Die Rechenoperation in der letzten Zeile von Tabelle 28 dient dazu, Werte zur Neuausrichtung des Roboterwerkzeugs für die anstehende Öffnungsbewegung um die Drehachse der Tür im nachfolgenden Roboterschnitt zu berechnen. Eine fortlaufende Neuberechnung ist notwen-

dig, da durch den variablen Parameter *[Winkel]* der Botschaft *Kreisbewegung* die Zielposition variieren kann, sie erfolgt auf Basis der über die Antwortbotschaft empfangenen Zielposition.

Zur Parametrierung eines Roboterschritts ist ein weiteres Grundgerüst unverzichtbar, mit dessen Hilfe der Bediener schrittweise die Parameterwerte der Botschaften vergibt oder konkrete Zielpositionen mit dem Roboterwerkzeug über das Handbediengerät anfährt. Das Grundgerüst zur Parametrierung des Roboterschritts *Tür entriegeln Roboter außen* ist in Tabelle 29 aufgeführt. Grundsätzlich ist das Parametrisieren von Grundgerüsten nur möglich, nachdem das vorgesehene Koordinatensystem, in diesem Beispiel das Türkoordinatensystem, festgelegt wurde.

Botschaft / Operation	Beschreibung und Parametrierung
<i>Bewegungsfreigabe</i>	Bewegungsfreigabe sofort entziehen
<i>Bedienerkommunikation</i>	Aufforderung zum Wechsel in den Handbetrieb des Roboters
<i>Überschleifen</i>	Überschleifen deaktivieren
<i>Werkzeugparameter einstellen</i>	Parameter des Roboterwerkzeugs zum Entriegeln übertragen und einstellen
<i>Bedienerkommunikation</i>	Aufforderung zum manuellen Positionieren des Werkzeugs
Bediener bewegt das Roboterwerkzeug manuell zu Position ②	
<i>Position auslesen</i>	Auslesen und Speichern der Position im Türkoordinatensystem
<i>Rechenoperation</i>	Ebene <i>Tür geschlossen</i> erzeugen
<i>Bedienerkommunikation</i>	Abfrage des Drehwinkels zur Ausgangsposition der Achse 2
<i>Bedienerkommunikation</i>	Abfrage der Strecke zur Einfahrt in den Türgriff
<i>Bedienerkommunikation</i>	Abfrage der Strecke an den Türgriff
<i>Bedienerkommunikation</i>	Abfrage des Radius des Türgriffs
<i>Bedienerkommunikation</i>	Abfrage der Strecke zum Entriegeln
<i>Bedienerkommunikation</i>	Abfrage der Mittelpunktverschiebung des Türgriff – Drehpunkts
<i>Bedienerkommunikation</i>	Aufforderung zum Wechsel in den Automatikmodus des Roboters
<i>Bedienerkommunikation</i>	Aufforderung zum Verlassen des Gefahrenbereichs und anschließenden Einschalten der Armleistung
<i>Achsenbewegung</i> ①	Drehung der Achse 2 zur Ausgangsposition
<i>Bewegungsfreigabe</i>	Bewegungsfreigabe erteilen und nach Bewegungsende entziehen
Roboter führt Bewegungsablauf ① aus	
<i>Position auslesen</i>	Auslesen der Ausgangsposition im Türkoordinatensystem
<i>Position anfahren</i> ②	Bewegung zur Position über dem Türgriff
<i>Linearbewegung</i> ③	Einfahrt in den Türgriff
<i>Linearbewegung</i> ④	Bewegung an den Türgriff
<i>Kreisbewegung</i> ⑤	Kreisbewegung am Türgriff zum Entriegeln der Tür
<i>Bewegungsfreigabe</i>	Bewegungsfreigabe erteilen und nach Bewegungsende entziehen
Roboter führt Bewegungsablauf ② bis ⑤ aus	
<i>Armleistung ausschalten</i>	Abschluss der Parametrierung und Freigabe des Gefahrenbereichs

Tabelle 29: Grundgerüst zur Parametrierung des Roboterschritts *Tür entriegeln Roboter außen*

Im Anschluss an die Parametrierung ist ein Testlauf mit automatisch verlangsamter Roboter- geschwindigkeit notwendig, um die Ausführung des Roboterschritts für Türbetätigungen frei- zugeben.

Der Programmcode für die Grundgerüste eines Roboterschritts sowie die dazugehörige Para- metrierung sind in einem separaten Fenster der Komponente *Robotik* hinterlegt. Abbildung 78 zeigt das Fenster für die Verwaltung der Methoden und Parameter zur Ausführung einer Tür- bewegung.

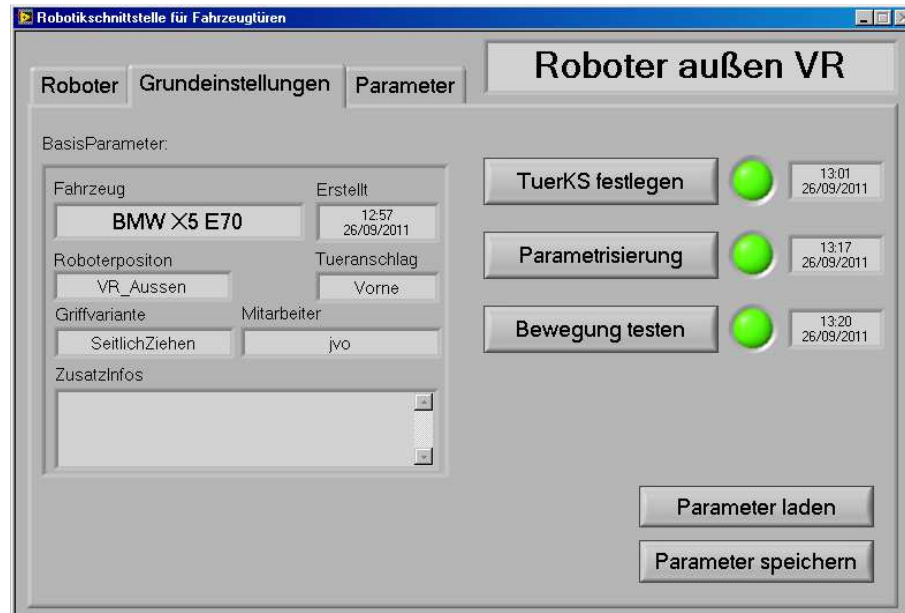


Abbildung 78: Bedienoberfläche zum Erzeugen einer Türbewegung

Die Parameter *Roboterposition*, *Türanschlag* und *Griffvariante* in Abbildung 78 sind in der Lage, Fehlerquellen bei der Parametrierung zu minimieren. Mit Hilfe der Eingaben für *Türan- schlag* und *Roboterposition* wird beispielsweise die Drehrichtung des Roboterarms beim Auf- schwenken oder Schließen einer Fahrzeugtür eindeutig festgelegt. Kollisionen durch Vorzei- chenfehler bei Winkelangaben sind dadurch ausgeschlossen.

Nach einem erfolgreich verlaufenden Test des Bewegungsablaufs ist eine Speicherung der Pa- rameter für eine künftige Wiederverwendung möglich. Für diese erneute Verwendung ist le- diglich das Türkoordinatensystem neu festzulegen. Neben der Einsparung von Positionen zu Gunsten von Parametern als Eingangswerte für einen Bewegungsablauf ist dies ein weiterer Aspekt zur Erhöhung der Reproduzierbarkeit von Prüfungen.

Die im Verlauf der Parametrierung eingegebenen Werte und die manuell angefahrenen Po- sitionen sind sichtbar auf der Bedienoberfläche hinterlegt. Abbildung 79 zeigt sämtliche einzugebenden Parameter für eine Türbewegung, vom Entriegeln bis zum Schließen der Tür. Die Werte für *Strecke zum Entriegeln*, *Winkel Tür in Zwischenposition* und *Winkel Tür Auf- schwenken* können über Offsets manuell oder automatisiert angepasst werden, um Verände- rungen am Prüfling auszugleichen.

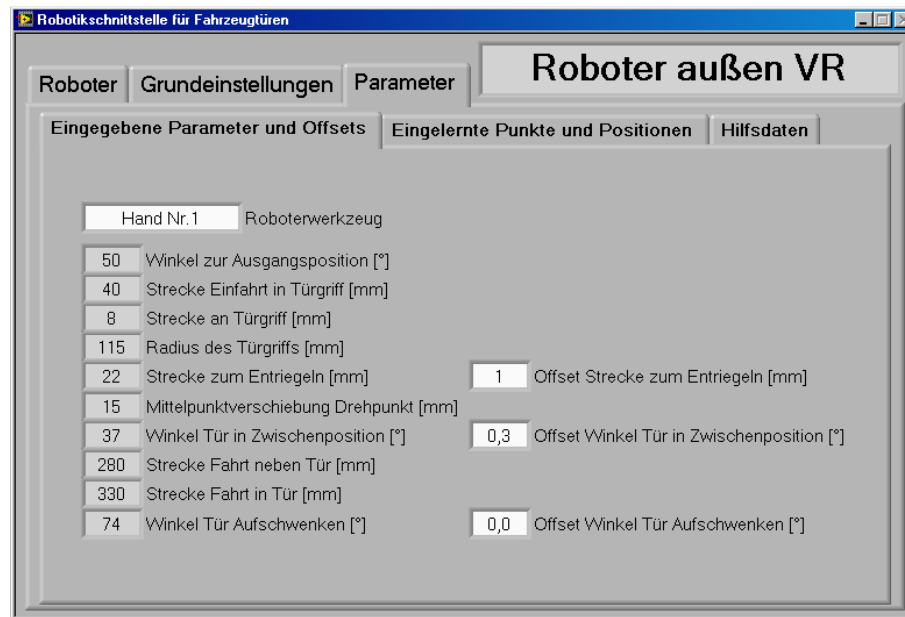


Abbildung 79: Bedienoberfläche zur Verwaltung der Parameter eines Grundgerüsts

Der Bewegungsablauf zum Ausführen einer Türbewegung auf Basis des herkömmlichen Teach – In – Verfahrens ist durch 21 Positionen definiert. Anpassungen im Bewegungsablauf sind ausschließlich durch das erneute Einlernen der betreffenden Positionen erreichbar. Die im Kontext der aufgabenorientierten Schnittstelle beschriebene Methode benötigt zusätzlich zum Türkoordinatensystem vier Positionen und zehn Parameter für die Ausführung einer Türbewegung. Erforderliche Anpassungen des Bewegungsablaufs während des Betriebs sind über die Vergabe von Offsets problemlos möglich. Für die Inbetriebnahme einer Türbewegung mit Hilfe der aufgabenorientierten Schnittstelle müssen maximal 30 Minuten eingeplant werden, daraus resultiert eine Zeitersparnis von zirka 90 Minuten gegenüber der herkömmlichen Methode.



## 5 Realisierung eines Prüfstands zur Dauererprobung von Fahrzeugtüren

Fahrzeugtüren sind sehr bedeutende Bestandteile der Karosserie. Sie bestehen aus einer Vielzahl mechanischer und elektronischer Komponenten und müssen den Insassen Komfort sowie zugleich Sicherheit bieten. Für den ersten Qualitätseindruck des Kunden sind das haptische Gefühl des Türgriffs sowie das Klangbild des Schließgeräuschs entscheidend, noch bevor Fahrverhalten oder Motorengeräusch beurteilt werden können [Lin-08].

Die im Rahmen einer Lebensdauererprobung relevanten Komponenten einer Tür sind Türrohbau, Scharniere, Türbremse, Dichtungen, Innenverkleidung, Schloss, Fensterheber und Innen- sowie Außengriff zum Entriegeln. Fahrzeuge der Oberklasse verfügen zusätzlich über Rollos, Türzuziehhilfe und Schließautomatik. Gegenläufig öffnende Türen, diese sind beispielsweise am *Rolls – Royce Ghost* des Baujahrs 2009 oder dem *Opel Meriva B* aus dem Jahr 2010 verbaut, benötigen zusätzliche Sicherheitsmechanismen, die ein Öffnen der Tür oberhalb der Schrittgeschwindigkeit blockieren.

Das in Kapitel 4 entwickelte Prüfsystemkonzept bietet eine Basis für alle erforderlichen Prüfstände zur realen Erprobung von Karosseriekomponenten. Kapitel 5 beschreibt weitere unerlässliche Hard- und Softwarekomponenten zur Lebensdauererprobung von Fahrzeugtüren sowie deren praktische Umsetzung. Seitens der Hardware werden zweckmäßig dimensionierte Roboterarme, Roboterwerkzeuge zur Innen- und Außenbetätigung von Türen sowie eine Adaption der Roboter an die Karosserie benötigt. Hauptbestandteil der softwareseitigen Ergänzungen sind die Prüfmodule. Diese beinhalten den Programmcode zur Automatisierung der Betätigungen samt Abstraktion in Roboterschritte, die Messdatenerfassung und Weiterverarbeitung sowie eine Bedienoberfläche zur Parametrierung von Bewegungsabläufen. Fehlen zur Umsetzung der Prüfmodule spezielle Eingangs- oder Ausgangsgrößen, so schaffen Erweiterungen des Auswertungs- und Ansteuerungskonfigurators, zusätzliche Umgebungsvariablen zur Kommunikation zwischen LabVIEW und CANoe oder Ergänzungen in der Parameter- sowie Messdatenverwaltung Abhilfe.

Die an Beispielen erläuterte Umsetzung des Prüfsystemkonzepts beschränkt sich im Rahmen dieser Arbeit auf Türen. Weitere bewegliche Karosseriekomponenten, wie Motorhaube, Heckklappe, Tankklappe oder Schiebedach, weisen keine grundsätzlich abweichenden Eigenarten auf, diese Bauteile lassen sich mit den Hard- und Softwarekomponenten zur Türerprobung in leicht modifizierter Form ebenfalls prüfen.

### 5.1 Prüfungsspezifische Hardware

#### 5.1.1 Roboterarme

Die Firma *Stäubli* bietet für den ausgewählten Robotercontroller CS8C Roboterarme mit diversen Reichweiten, Traglasten und Winkelgeschwindigkeiten der Achsen an. Zur Erprobung von Fahrzeugtüren eignen sich die Roboterarme *TX60L* zur Türinnenbetätigung und *RX160L* zur Türaußenbetätigung.





		
	Türinnenbetätigung	Türaußenbetätigung
M	TX60L [Stä-11b]	RX160L [Stä-11a]
R	920mm	2010mm
L/E	5kg / 53kg	28kg / 250kg
W4	$\pm 270^\circ$	$\pm 270^\circ$
G1-3	435°/s, 385°/s, 500°/s	200°/s, 200°/s, 255°/s

Tabelle 30: Ausgewählte Roboterarme zur Türbetätigung

Legende: M: Modell, R: Reichweite, L: Traglast, E: Eigengewicht, W4: Maximaler Drehwinkel der Achse 4, G1-3: Maximale Winkelgeschwindigkeiten der Achsen 1 – 3.

Der Türaußengriff als Kraftangriffspunkt zum Aufschwenken und Schließen einer Tür befindet sich im Vergleich zum Türinnengriff weiter von der Scharnierachse entfernt. Da mit zunehmender Größe der Robotermodelle die maximale Winkelgeschwindigkeit der Achsen abnimmt, ist für die Außenbetätigung eine optimale Positionierung des Roboterarms zur Tür erforderlich. Andernfalls können die geforderten Öffnungs- und Schließgeschwindigkeiten nicht erreicht werden. Die Achse 1 des Roboters muss zu diesem Zweck möglichst nah an der Drehachse der Tür positioniert sein. Ebenso gelten zur Vermeidung von Kollisionen mit übrigen Karosserieelementen beim Schließvorgang weitestgehend identische Proportionen des *Oberarms* (Bereich zwischen Achse 2 und 3) und *Unterarms* (Abschnitt von Achse 3 bis 5) als Voraussetzung.

Bei dem Roboterarm zur Türinnenbetätigung ist das Verhältnis von Reichweite und Aufbauhöhe entscheidend. Aus der Fahrzeugmitte müssen die beiden Vorder- oder Hintertüren erreichbar sein, ohne dabei eine Kollision mit dem Fahrzeughimmel oder der Frontscheibe zu verursachen. Der Roboterarm *TX60L* misst vom Boden des Sockels bis zu seinem höchsten Punkt in der abgebildeten Armstellung aus Tabelle 30 820 Millimeter. Dies hat sich in der Praxis als oberste Grenze für Kleinwagen erwiesen. Reichweitendefizite bei Oberklassefahrzeugen gleicht das Roboterwerkzeug aus. Durch den üblicherweise nah an der Scharnierachse positionierten Zuziehgriff ist die Realisierung der geforderten Betätigungsgeschwindigkeiten unkritisch.

### 5.1.2 Roboterwerkzeuge

Die in Abschnitt 3.3.2 beschriebenen Ansätze für humanoide Greifwerkzeuge, besonders die am *DLR* entwickelte Roboterhand, sind sehr vielversprechend. Zur Dauererprobung von Karosserieelementen fehlt jedoch die Robustheit, besonders für den Einsatz unter extremen Klimabedingungen und bei großen Kräften. Ebenso übersteigen die Anschaffungs- und Instandhaltungskosten das Budget für Türversuche erheblich. Aus den Prüfvorschriften zur Türerprobung sowie den Anforderungen an das Prüfsystemkonzept leiten sich bezüglich der Roboter-

werkzeuge einige Erfordernisse ab: Die Roboterwerkzeuge müssen so robust ausgelegt sein, dass sie den mechanischen Belastungen durch die Türbewegung dauerhaft standhalten und unter extremen Umweltbedingungen zuverlässig funktionieren. Zur Absorption von Schlägen auf den Roboterarm beim Aufschwenken oder Schließen der Türen ist die Verwendung von Feder – Dämpfer – Systemen unverzichtbar. Für die Messung von Kräften beim Entriegeln am Türgriff sowie zur Berechnung der Schließenergie muss ein Kraftsensor integriert sein. Dabei ist eine sensorgerechte Einbaulage zu berücksichtigen. Eine Türbetätigung in realitätsnaher Form beinhaltet, dass keine Zusatzmassen am Prüfling befestigt werden und keine starre Verbindung zwischen dem Roboterwerkzeug und der Tür besteht. Ein simpler Aufbau sowie die Verwendung von Normteilen minimieren Kosten, Fertigungsdauer und Reparaturzeiten. Der Roboter für den Fahrzeuginnenraum verfügt mit fünf Kilogramm über eine vergleichsweise geringe Traglast. Das Roboterwerkzeug zur Türinnenbetätigung ist daher gewichtsoptimiert auszuführen. Potentielle Schwachpunkte, wie Schraubverbindungen, sind bei der Entwicklung zu vermeiden. Im Verlauf dieser Arbeit wurden in drei Iterationsschritten die nun vorliegenden Roboterwerkzeuge zur Türinnen- und Türaußenbetätigung entwickelt [Wey-11].

#### 5.1.2.1 Roboterwerkzeug zur Türaußenbetätigung

Abbildung 80 zeigt das Roboterwerkzeug zur Türaußenbetätigung mit den Hauptfunktionsträgern. Die Masse von 1,6 Kilogramm liegt weit unter der maximalen Traglast des Roboters. Sämtliche Bestandteile sind aus korrosionsbeständigen Werkstoffen gefertigt. Die Einstellung des Feder – Dämpfer – Systems ist in Abhängigkeit der Masse der zu bewegenden Tür variabel. Der Kraftsensor ist geschützt im Innern des Werkzeugs als Pendelstütze untergebracht, dies verhindert eine Übertragung von kritischen Biegemomenten auf die Kraftmessdose.

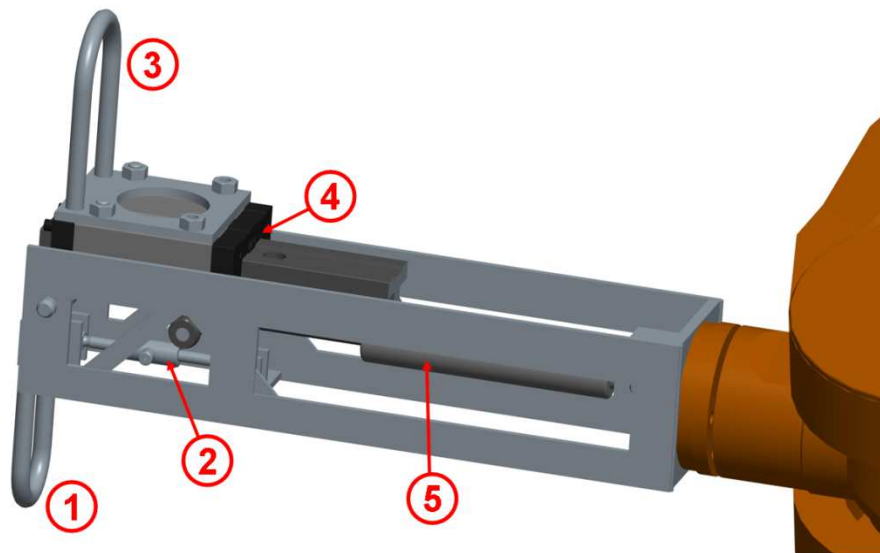


Abbildung 80: Roboterwerkzeug zur Türaußenbetätigung

①	Finger zum Entriegeln des Türaußengriffs und zum Schließen der Tür
②	Sensorelement zur Messung der Kraft am drehbar gelagerten Finger
③	GrößenvARIABLE <i>Mitnehmer</i> zum Aufschwenken der Tür
④	Linearführung für die Bewegung des Mitnehmers
⑤	Feder – Dämpfer – System zur Absorption von Schlägen auf den Mitnehmer

Tabelle 31: Hauptfunktionsträger des Roboterwerkzeugs zur Türaußenbetätigung

Aufgrund des drehbar gelagerten *Fingers* und des dadurch entstandenen Hebelarms entspricht der gemessene Kraftwert nicht der tatsächlich anliegenden Größe. Eine Umrechnung erfolgt direkt auf dem Robotercontroller. Im Parameter [*KraftFaktor*] der Botschaft *Werkzeugparameter einstellen* findet diese konstruktiv bedingte Eigenschaft Berücksichtigung.

### 5.1.2.2 Roboterwerkzeug zur Türinnenbetätigung

Abbildung 81 zeigt das Roboterwerkzeug zur Türinnenbetätigung sowie dessen Hauptfunktionsträger. Die Gesamtmasse summiert sich auf 1,45 Kilogramm und beträgt somit weniger als ein Drittel der maximalen Traglast des Roboters. Überdies ist die Massenträgheit der Tür beim Aufschwenken und Schließen als Belastung auf den Roboterarm einzukalkulieren. Üblicherweise verwendete Werte zwischen 20 und 40 Prozent für Geschwindigkeit und Beschleunigung haben in der Praxis nicht zu Ausfällen aufgrund einer Überlastung des Roboterarms geführt. Auch die Überwachung der Achsmomente mittels der Botschaft *Achsmomentexport* zeigte keine Überbelastung. Sämtliche Bestandteile des Roboterwerkzeugs sind aus korrosionsbeständigen Werkstoffen gefertigt. Die variable Einstellung des Feder – Dämpfer – Systems ist abhängig vom Kraftangriffspunkt beim Aufschwenken sowie der Masse der Tür. Die Kraftmessdose ist an einer geschützten Stelle als Pendelstütze installiert. Im Unterschied zum Roboterwerkzeug zur Außenbetätigung sind *Finger* und *Mitnehmer* über eine Vierkantschiene federgedämpft mit dem Roboter verbunden. Die Position des Fingers zum Roboter verschiebt sich nur bei einer Druckbelastung, beim zugartigen Entriegeln des Türinnengriffs ist somit eine korrekte Positionsmessung möglich. Die Länge der Vierkantschiene im Innern des Zylinders lässt sich in Abhängigkeit vom Fahrzeugmodell und der Roboterposition im Innenraum je nach benötigter Reichweite variieren.

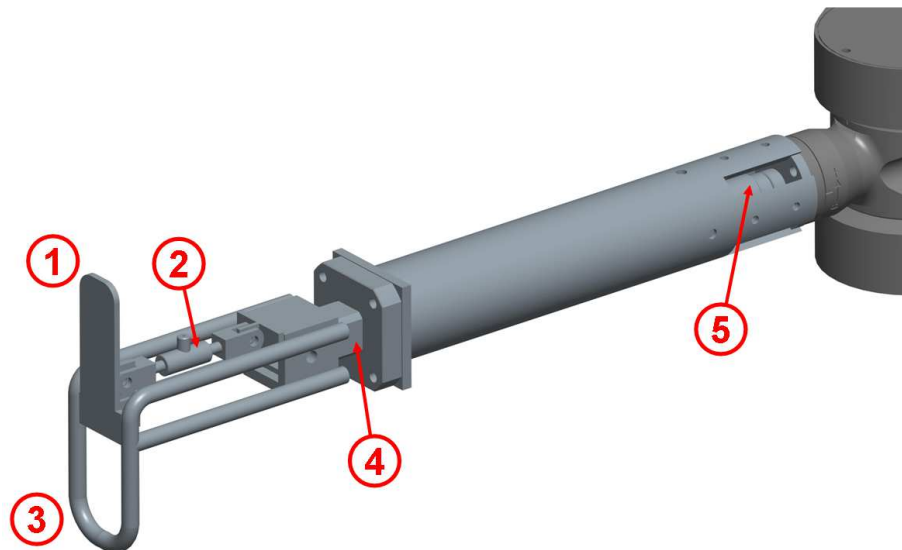


Abbildung 81: Roboterwerkzeug zur Türinnenbetätigung

①	<i>Finger</i> zum Entriegeln des Türinnengriffs
②	Sensorelement zur Messung der Kraft am drehbar gelagerten Finger
③	<i>Mitnehmer</i> zum Aufschwenken und Schließen der Tür
④	Vierkantschiene für die Linearbewegung von Finger und Mitnehmer
⑤	Feder – Dämpfer – System zur Absorption von Schlägen auf Finger und Mitnehmer

Tabelle 32: Hauptfunktionsträger des Roboterwerkzeugs zur Türinnenbetätigung

Ebenso wie beim Werkzeug zur Türaußenbetätigung ist eine Korrektur des gemessenen Kraftwerts am Finger aufgrund der Hebelverhältnisse notwendig.

### 5.1.3 Adaption der Roboter am Fahrzeug

Ein zeitintensiver Bestandteil der Inbetriebnahme einer Türerprobung ist der Aufbau der Hardwarekomponenten. Im Rahmen dieser Arbeit wurde ein Konzept zur Adaption der Roboter am und im Fahrzeug entwickelt. Dieses beinhaltet ein Podest für den Roboter zur Außenbetätigung, einen karosserieabhängigen Profilrahmen und ein Gestell zur Befestigung des Roboters im Fahrzeuginnenraum [Woi-11].

Das Roboterpodest zur Türaußenbetätigung hat einige Anforderungen zu erfüllen, um einen flexiblen und effizienten Einsatz der Roboter bei Türerprobungen zu erreichen. Primär muss ein stabiler und sicherer Stand des Roboters gewährleistet sein. Dies wird durch ein Eigengewicht des Podests von zirka 100 Kilogramm, durch höhenverstellbare Maschinenfüße mit Antirutschbeschichtung, selbsthemmende Schnellverschlüsse zur Arretierung an einer Profilschiene und schwenkbare Ausleger erreicht. Die Podesthöhe ist mit 400 Millimetern so gewählt, dass die Achse 2 des Roboters in 970 Millimetern Höhe mit der Durchschnittshöhe von Türaußengriffen übereinstimmt. Zur Ermittlung dieses Durchschnittswerts erfolgte eine Höhenmessung des Türaußengriffs von Gesamtfahrzeugen und Rohkarosserien auf Lackiertraversen vom Kleinwagen *Mini R56* (810 Millimeter) bis hin zur Luxuslimousine *Rolls – Royce Ghost* (1120 Millimeter auf einer Lackiertraverse). Die Angleichung der Höhenverhältnisse von Türaußengriff und Achse 2 bewirkt, dass sich bei gleichen Längenverhältnissen von Ober- und Unterarm der Arbeitsraum des Roboters nicht mit dem Bewegungsbereich der Tür überschneidet. Die Grundfläche des Podests ist mit  $600 \times 700$  Millimetern so gewählt, dass der Roboter ausreichend nah an der Drehachse der Tür positioniert werden kann. Dadurch wird ein optimaler Hebelarm in Bezug zur Achse 1 des Roboters gewährleistet, welcher zum Erreichen der geforderten Geschwindigkeiten wesentlich ist. Aufgrund der Anordnung der Maschinenfüße ist ein Transport mit dem Gabelstapler oder Hubwagen möglich. Sämtliche Bestandteile des Podests bestehen aus korrosionsbeständigen Werkstoffen.

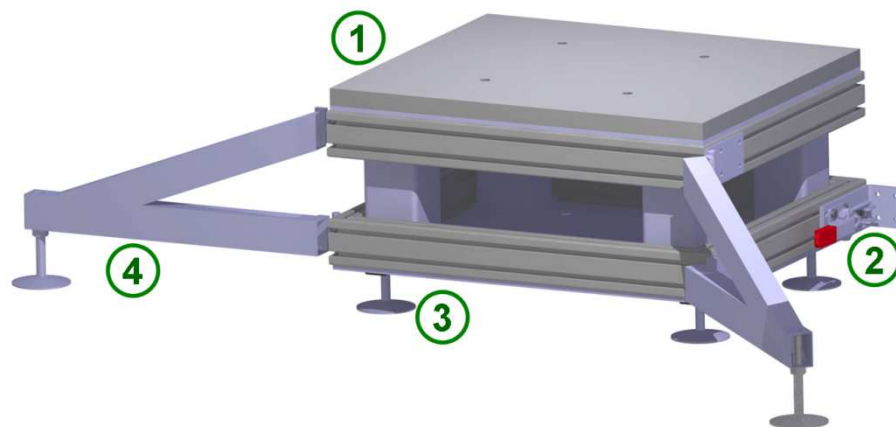


Abbildung 82: Podest für den Roboter zur Türaußenbetätigung

①	Massive Podestplatte für Stabilität und Eigengewicht (10mm Stärke)
②	Selbsthemmender Schnellverschluss
③	Höhenverstellbare Maschinenfüße mit Antirutschbeschichtung
④	Variabler Ausleger zur Abstützung im Arbeitsbereich

Tabelle 33: Hauptfunktionsträger des Roboterpodests zur Türaußenbetätigung

Bei der Ausführung des Profilrahmens zur Adaption des Roboterpodests an die Karosserie wird zwischen vollständig ausgestatteten Komplettfahrzeugen und Rohkarosserien differenziert, welche auf Lackiertraversen befestigt sind. Bei Komplettfahrzeugen kommt pro Fahrzeugseite eine Profilschiene zum Einsatz. Diese ist an zwei mit Antirutschmatten bestückten Bodenplatten verschraubt, welche durch das Eigengewicht des Fahrzeugs fest mit dem Halbenboden verbunden werden und so einen stabilen Befestigungssockel für die Roboterpodeste bilden. Die Fahrzeugreifen sind über eine Spannvorrichtung auf den Bodenplatten verkeilt, deren stufenlose Verschiebung an der Profilschiene möglich ist. Der passende Radstand ist mit wenigen Handgriffen einstellbar.

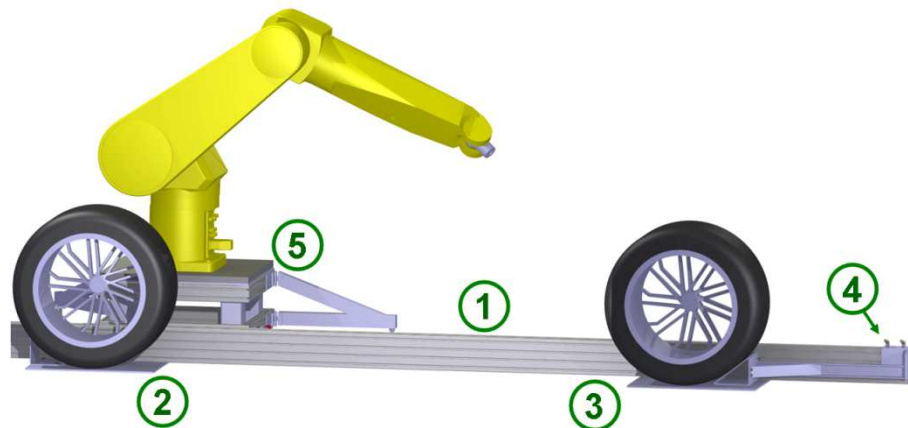


Abbildung 83: Profilschiene zur Adaption der Roboter an Komplettfahrzeuge

①	Profilschiene zur Verschraubung der Schnellverschlüsse des Podests und Bodenplatten
②	Bodenplatte mit festem Keil als Unterlage für das Vorderrad
③	Bodenplatte mit verschiebbarem Keil als Unterlage für das Hinterrad
④	Spannvorrichtung zur Befestigung des Keils am Hinterrad
⑤	Podest für den Roboter zur Türaußenbetätigung

Tabelle 34: Hauptfunktionsträger der Profilschiene für Komplettfahrzeuge

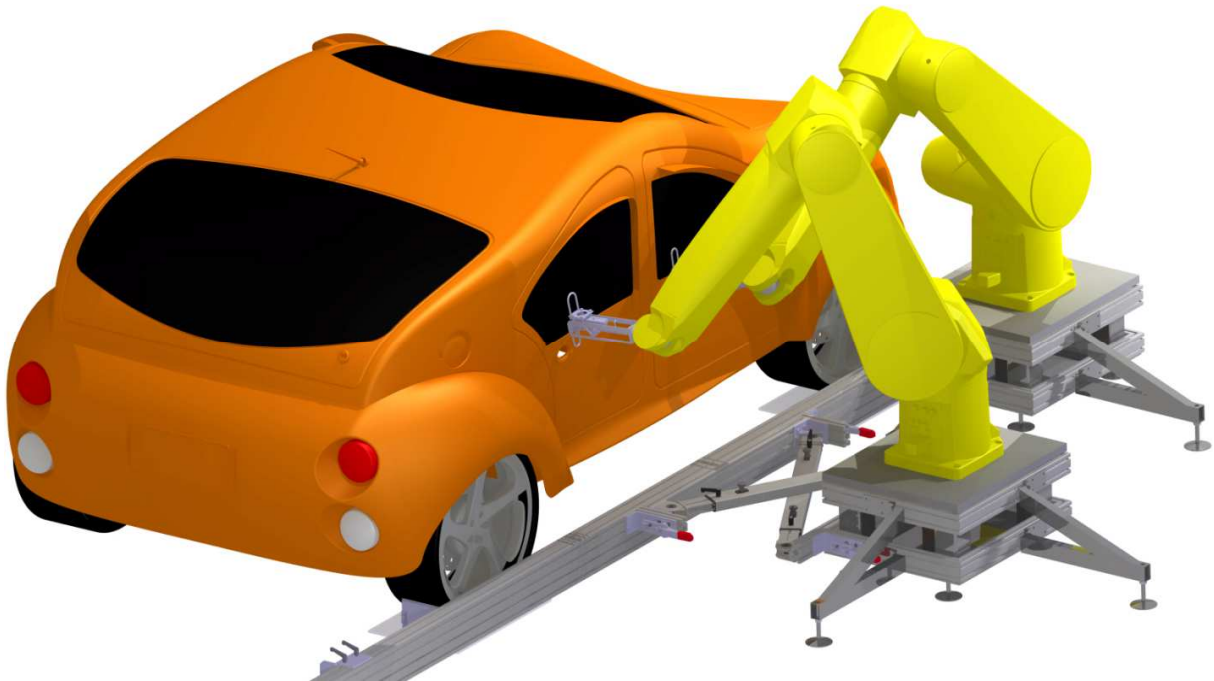
Bei Rohkarosserien ist eine weniger aufwendige Konstruktion verwendbar, da der Abstand zwischen der vorderen und hinteren Lackiertraverse bei allen Fahrzeugtypen eines Herstellers gleichbleibend ist. Zur Adaption der Roboterpodeste genügt ein rechteckiger Rahmen aus Profilschienen, welcher mit Aufnahmen für die Bolzen der Lackiertraversen ausgestattet ist. Zum Transport empfehlen sich Rollen, die bei Bedarf unterhalb des Rahmens verschraubt werden können.



Abbildung 84: Grundrahmen zur Adaption der Roboter an Rohkarosserien



Ist im Prüfablauf vorgesehen, die vordere und hintere Tür einer Fahrzeugseite parallel zu testen, muss der Roboter zur Außenbetätigung der hinteren Tür mit einem gewissen Abstand zur Profilschiene montiert sein. Bei einer Positionierung in optimaler Stellung zur Drehachse der hinteren Tür direkt an der Profilschiene würde sich der Roboter unmittelbar im Bewegungsbereich der vorderen Tür befinden. Abbildung 85 zeigt den Aufbau, bei dem durch ein breitenvariables Gestell eine optimale Anordnung von zwei Robotern an einer Fahrzeugseite möglich ist. Die Adaption des Gestells geschieht mit baugleichen Schnellverschlüssen wie beim Podest.



**Abbildung 85: Prüfaufbau zur Betätigung beider Türen an einer Fahrzeugseite**

In Abbildung 85 werden die sehr engen Platzverhältnisse bei einer parallelen Betätigung zweier Türen einer Fahrzeugseite von außen deutlich. Trotzdem ist es mit etwas Geschick und Erfahrung in der Praxis möglich, die Roboter für einen kollisionsfreien Ablauf zu positionieren. Durch die fehlende Nähe zur Drehachse ist der Roboteraufbau an der hinteren Tür in Bezug auf die Öffnungs- und Schließgeschwindigkeiten nicht optimal. Tests haben jedoch bestätigt, dass auch in dieser Konstellation Schließgeschwindigkeiten von zwei Metern pro Sekunde auf Höhe des Türschlosses erreichbar sind. Bei Fahrzeugen mit gegenläufig öffnenden Türen besteht diese Problematik nicht.

Zur Befestigung der Roboter im Fahrzeuginnenraum eignen sich optimal die Anschraubpunkte der Vordersitze, der Rücksitzbank oder der Verzurrschienen im Kofferraum. Diese Punkte sind für die Aufnahme von großen Kräften ausgelegt und in einer ausreichenden Anzahl im Innenraum vorhanden. Die Konstruktion eines Universalgestells hat sich als nicht zweckmäßig herausgestellt, da die engen Platzverhältnisse, die Höhe und Breite des Getriebekanals sowie der Abstand zwischen den Vordersitzen je nach Fahrzeugtyp stark variieren. Sollen bei einem heckgetriebenen Fahrzeug beide Vordertüren parallel betätigt werden, ist häufig der Getriebekanal zu entfernen. Diese Möglichkeit ist aber nur bei Rohkarosserien gegeben, bei Komplettfahrzeugen wäre der Aufwand zu groß. In diesem Fall bleibt nur die zeitintensivere Variante, beide Türen nacheinander zu erproben. Abbildung 86 zeigt ein Gestell aus Profilschienen an der Stelle des Fahrersitzes mit dem Sockel des *TX60L*.





Abbildung 86: Gestell zur Adaption des Roboters im Fahrzeuginnenraum

Die in diesem Kapitel dargestellten Konstruktionen machen am Beispiel des Prüfstands zur Türerprobung deutlich, in welcher Weise Forderungen nach einem ortflexiblen Einsatz bei gleichzeitig stabiler Adaption der Roboterarme umsetzbar sind. Der Wunsch nach einer Verwendung von Normbauteilen ist weitestgehend berücksichtigt, Spezialanfertigungen lassen sich aber nicht gänzlich vermeiden.

## 5.2 Prüfmodule zur Türerprobung

Aus der Anzahl von mechanischen und elektronischen Komponenten einer Fahrzeugtür ergeben sich gleichermaßen viele Prüfmodule. Konkret sind zur Türerprobung die Prüfmodule *Türbewegung*, *Türschlossbetätigung*, *Fensterheberbetätigung*, *Rollobetätigung*, *Außenspiegelbetätigung* und *Türzuziehhilfenbetätigung* in der Prüfablaufverwaltung verfügbar. Weitere allgemeine Prüfmodule, wie *Pause*, *Lange Pause* und *Diagnosefunktion* werden ebenso bei Türerprobungen eingesetzt.

Ein Teil des Programmcodes der Prüfmodule kann aus universell verwendbaren *Basisprogrammen* generiert werden. Dazu gehört die Abfrage von Parameterwerten, das Ermitteln des Prüflingsstatus, das Erzeugen einer Ansteueranweisung oder die Erfassung von Messdaten. Der zweite Teil, ein sehr individueller Programmcode, ist aufgrund von Prüfvorschriften für das Bauteil, zur Erkennung und Dokumentation von Fehlfunktionen und zur Sicherung der Laufstabilität der Prüfung individuell anzupassen und zu optimieren.

In diesem Kapitel wird die interne Struktur von vier typischen Prüfmodulen zur Erprobung von Fahrzeugtüren dargestellt. So beinhaltet *Türbewegung* eine überwiegend mechanische Betätigung durch Roboter, *Türschlossbetätigung* ist eine Kombination aus elektronischer Ansteuerung und mechanischer Überprüfung. *Fensterheberbetätigung* beschreibt eine rein elektronische Ansteuerung und Auswertung, *Diagnosefunktion* steht beispielhaft für eine Überwachungsvariante des Prüflings während der Lebensdauerprüfung.

Die Tabellen 35 bis 37 machen zeilenweise die einzelnen Zustände des Zustandsautomaten deutlich, auf denen der interne Ablauf eines Prüfmoduls basiert. Blau unterlegte Zustände zeigen den wesentlichen Teil der Prüfung an, gelb markierte Zustände dienen der Anzeige der Organisation oder Kommunikation mit parallel laufenden Prüfmodulen. Die farbig markierten Pfeile geben mögliche Abarbeitungsreihenfolgen der einzelnen Zustände vor.

### 5.2.1 Türbewegung

Das Prüfmodul *Türbewegung* dient hauptsächlich der mechanischen Absicherung von Türrohbau, Türbremse, Türaußen- und Türinnengriff samt der Schlossanbindung, Türinnenverkleidung und Teilen des Dichtungssystems.

Der mittels eines Roboters ausgeführte Bewegungsablauf beginnt mit dem Entriegeln der Tür am Türaußen- oder Türinnengriff. Bei einer Außenbetätigung folgt nun eine Bewegung der Tür zur ersten Zwischenraste, um das Roboterwerkzeug zum Aufschwenken an der Türinnen-seite zu orientieren. Diese Neuorientierung ist bei einer Innenbetätigung nicht notwendig, es wird lediglich eine Bewegung zum Kraftangriffspunkt an der Türinnenverkleidung durchgeführt. Im Anschluss erfolgt das Aufschwenken der Tür bis in die Endraste. Der Mitnehmer am Roboterwerkzeug dämpft dabei mögliche Rückstöße auf den Roboterarm und sorgt für ein Einpendeln der Tür in die Endraste. Abschließend wird die Tür am Außengriff, oder bei einer Betätigung von innen am Türzuziehgriff, geschlossen. Abbildung 87 beschreibt in einer Bilderserie beispielhaft den beschriebenen Vorgang für die Betätigung von außen.



Abbildung 87: Bilderserie des Prüfmoduls *Türbewegung*

Der in Abbildung 87 dargestellte Bewegungsablauf ist programmtechnisch in die vier Roboterschritte *Tür entriegeln*, *Tür in Zwischenposition*, *Tür aufschwenken* und *Tür schließen* unterteilt. Tabelle 35 beschreibt deren Abfolge sowie parallel ausgeführte Statusabfragen und Messungen.

Zustand		Bemerkungen	Ansteuerbefehle, Meldungen	Statusabfragen, Messungen	Zielzustand
1	INIT	Prüfmodulparameter auswerten	Meldung an Prüfablaufverwaltung	-	●→
2	FREIG?	Prüfmodulausführung oder Roboter gesperrt?	-	Prüfmodulfreigabe (T) Roboterfreigabe (T)	●→
3	AKTIV!	Prüfmodulstatus aktiv setzen, Roboter (Innenbetätigung) sperren	Prüfmodul-, Roboterstatus in Datenbasis setzen	-	●→
4	STATUS TÜR?	Tür geschlossen?	-	Türöffnung (T)	●→
5	ENTRIE!	Tür entriegeln	Roboterschritt <i>Tuer entriegeln</i>	Türöffnung (T) Kraft (M), Weg (M) Türöffnungswinkel (M) Auslöseweg (C)	●→
6	ZWISCH!	Tür in Zwischenposition bewegen	Roboterschritt <i>Tuer in Zwischenposition</i>	Türöffnungswinkel (M) Winkelgeschwindigkeit (M) Rastwinkel (C)	●→
7	AUF SCH!	Tür aufschwenken	Roboterschritt <i>Tuer aufschwenken</i>	Türöffnungswinkel (M) Winkelgeschwindigkeit (M) Öffnungsgeschwindigkeit (C) Rastwinkel (C)	●→
8	SCHLIES!	Tür schließen	Roboterschritt <i>Tuer schliessen</i>	Türöffnung (T), Kraft (M), Weg (M) Türöffnungswinkel (M) Winkelgeschwindigkeit (M) Schließgeschwindigkeit (C) Schließenergie (C)	●→
9	FREIG!	Roboter (Innenbetätigung) freigeben	Meldung an Datenbasis	-	●→
10	IO	Fehlerfreie Ausführung des Prüfmoduls	Meldung an Prüfablaufverwaltung	-	●→
11	FEHLER	Fehler bei der Ausführung des Prüfmoduls	Fehlermeldung an die Ereignisanzeige	-	●→
12	ENDE	Ende des Prüfmoduls, Status auf inaktiv setzen	Meldung an Datenbasis und Prüfablaufverwaltung	-	●→

T: Abfrage mit Timeout, M: Messung, C: Charakteristischer Wert  
 ●→: Ablauf ohne Fehler, ●→: Ablauf bei Fehler, ●→: Ablauf fehlerunabhängig

Tabelle 35: Interner Ablauf des Prüfmoduls *Türbewegung*

Im Zustand *INIT* werden die Prüfmodulparameter ausgewertet, die in einem direkten Zusammenhang mit der Position des Prüfmoduls im Prüfplan stehen. Hier heißen diese *ausführender Roboter* (Innenbetätigung, Außenbetätigung) sowie *Ausführungshäufigkeit* (jeder Zyklus, jeder 2. Zyklus, ...).

Zu den erhaltenen Daten der Statusabfragen und Messungen der Zustände 4 bis 8 sind in der Parameterverwaltung entsprechende Grenzwerte, beispielsweise *Timeout Ermittlung Status Türöffnung*, eingetragen. Ein Über- oder Unterschreiten dieser Grenzwerte bedeutet einen direkten Sprung in den Zustand *Fehler*. Das Prüfmodul wird abgebrochen und die Dauererprobung pausiert. In der überwiegenden Anzahl dieser Fälle liegen mechanische Defekte an der Tür vor, welche genauer zu analysieren sind.

Auf Basis der Messkurven von Kraft und Weg im Zustand *ENTRIE!* kann der benötigte Auslöseweg zum Entriegeln der Tür als charakteristischer Wert bestimmt werden. Dieser entspricht dem zurückgelegten Weg an der Position des Kraftmaximums. Erhöht sich der Auslöseweg mit zunehmender Erprobungsdauer durch eine Verlängerung des Bowdenzugs, muss zur Sicherstellung der Laufstabilität der Parameter *Strecke entriegeln* um eine entsprechende Anzahl von Millimetern erhöht werden.

### 5.2.2 Türschlossbetätigung

Das Prüfmodul *Türschlossbetätigung* ist eine Kombination aus mechanischer und elektronischer Prüfung, der Inhalt ist die Kontrolle der elektronisch ausgeführten Verriegelung des Türschlosses durch Ziehen am Türaußengriff oder am Türinnenöffner. In erster Linie wird die Zuverlässigkeit des sicherheitsrelevanten Verriegelungsmechanismus abgesichert.

Zustand		Bemerkungen	Ansteuerbefehle, Meldungen	Statusabfragen, Messungen	Zielzustand
1	INIT	Prüfmodulparameter auswerten	Meldung an Prüfablaufverwaltung	-	●
2	AKTIV!	Prüfmodulstatus aktiv setzen	Prüfmodulstatus in Datenbasis setzen	-	●
3	STATUS TÜR?	Tür geschlossen?	-	Türöffnung (T)	●
4	VERRIE!	Türschloss verriegeln	Türschloss verriegeln	Schlossstatus (T) Strom (M), Spannung (M) Betätigungsenergie (C)	●
5	KONTZ!	Kontrollziehen am Türaußengriff	Roboterschritt <i>Kontrollziehen Hinfahrt</i>	Kraft (M), Weg (M)	●
6	STATUS TÜR?	Kontrolle, ob Tür geschlossen?	-	Türöffnung (T)	●
7	HOME!	Roboterbewegung auf Homeposition	Roboterschritt <i>Kontrollziehen Rueckfahrt</i>	Kraft (M), Weg (M)	●
8	ENTR!	Türschloss entriegeln	Türschloss entriegeln	Schlossstatus (T) Strom (M), Spannung (M) Betätigungsenergie (C)	●
9	IO	Fehlerfreie Ausführung des Prüfmoduls	Meldung an Prüfablaufverwaltung	-	●
10	FEHLER	Fehler bei der Ausführung des Prüfmoduls	Fehlermeldung an die Ereignisanzeige	-	●
11	ENDE	Ende des Prüfmoduls, Status auf inaktiv setzen	Meldung an Datenbasis und Prüfablaufverwaltung	-	●

T: Abfrage mit Timeout, M: Messung, C: Charakteristischer Wert  
 ●→: Ablauf ohne Fehler, ●→: Ablauf bei Fehler, ●→: Ablauf fehlerunabhängig

**Tabelle 36: Interner Ablauf des Prüfmoduls *Türschlossbetätigung***

Der Verlauf der Betätigungsenergie beim Ver- oder Entriegeln über die Lebensdauererprobung sowie die Einflüsse extremer Umweltbedingungen, berechnet in den Zuständen 4 und 8, sind funktionsrelevante Größen. Typischerweise werden die Antriebe der Schlösser aller Türen über eine gemeinsame Sicherung versorgt. Sollte diese schmelzen, kann das Fahrzeug nur durch eine Notentriegelung an der Fahrertür betreten und verlassen werden. Eine derartige Situation verursacht beim Kunden eine große Unzufriedenheit. Anhand des Verlaufs der charakteristischen Werte lässt sich beurteilen, ob unter dem Einfluss einer Alterung das Verhältnis zwischen Sicherung und Stromaufnahme ausreichend dimensioniert ist.

Das dargestellte Prüfmodul arbeitet unabhängig von der Fahrzeugelektronik, da üblicherweise eine Ansteuerung einzelner Schlossantriebe im Rahmen der Zentralverriegelung durch das Steuergerät nicht vorgesehen ist. Für die unabhängige Prüfung jeder einzelnen Tür ist daher eine externe Ansteuerung der Antriebe über die Relais des Prüfstandsrechners unerlässlich. Diese Methode ist nicht uneingeschränkt realitätsgetreu, sie stellt aber den besten Kompromiss dar.





Beim Auftreten eines Fehlers in einem Zustand wird das Prüfmodul abgebrochen und die Dauererprobung pausiert. In diesem Fall liegt mit hoher Wahrscheinlichkeit ein mechanischer Defekt vor, der analysiert werden muss.



### 5.2.3 Fensterheberbetätigung

Die Ansteuerung im Prüfmodul *Fensterheberbetätigung* erfolgt ausschließlich elektronisch. Der Fokus der Erprobung ist auf die Mechanik des Fensterhebersystems und die angrenzenden Dichtungen gerichtet. Weiterhin relevant ist die optische und akustische Güte der Fahrbewegung der Fensterscheibe. Einbezogen in die Absicherung sind ferner Parameter im Steuergerät, beispielsweise des sicherheitsrelevanten Einklemmschutzes.

Zustand		Bemerkungen	Ansteuerbefehle, Meldungen	Statusabfragen, Messungen	Zielzustand
1	INIT	Prüfmodulparameter auswerten	Meldung an Prüfablaufverwaltung	-	↓
2	CAN LOG?	Datenaufzeichnung von CAN – Botschaften durchführen?	-	-	↓
3	SPERR!	Übrige Prüfmodule für CAN – Datenaufzeichnung sperren	Meldung an Datenbasis	-	↓
4	ABGES?	Warten bis sämtliche Prüfmodule abgeschlossen	-	Abfrage Status in Datenbasis (T)	↓
5	START LOG!	CAN – Datenaufzeichnung starten	Meldung über Datenbasis an CANoe	CAN – Botschaften (M) (bis Zustand <i>STOPP LOG!</i> )	↓
6	AKTIV!	Prüfmodulstatus aktiv setzen	Prüfmodulstatus in Datenbasis setzen	-	↓
7	FH BETAET!	Fensterheberbetätigung	Übertragung der Fensterheberbetätigung zum Ansteuerungskonfigurator	Fensterheberantrieb (T) Fensterheberstatus (T) Fensterheberfehler (M) Strom (M), Spannung (M) Betätigungsenergie (C)	↓
8	STOPP LOG!	CAN – Datenaufzeichnung stoppen	Meldung über Datenbasis an CANoe	-	↓
9	IO	Fehlerfreie Ausführung des Prüfmoduls	Meldung an Prüfablaufverwaltung	-	↓
10	FEHLER	Fehler bei der Ausführung des Prüfmoduls	Fehlermeldung an die Ereignisanzeige	-	↓
11	ENDE	Ende des Prüfmoduls, Status auf inaktiv setzen	Meldung an Datenbasis und Prüfablaufverwaltung	-	↓

T: Abfrage mit Timeout, M: Messung, C: Charakteristischer Wert  
: Ablauf ohne CAN – Datenaufzeichnung, : Ablauf mit CAN – Datenaufzeichnung, : Ablauf bei Fehler, : Ablauf unabhängig von CAN – Datenaufzeichnung

**Tabelle 37: Interner Ablauf des Prüfmoduls *Fensterheberbetätigung***

Die sorgfältige Abstimmung der Parameter für den Einklemmschutz hat einen hohen Stellenwert. Auf der einen Seite muss die Bewegungsumkehr der Fensterscheibe im Ernstfall, ein sogenannter *Reversierer*, uneingeschränkt funktionieren. Andererseits dürfen mit zunehmenden Fahrzeugalter keine Fehlfunktionen auftreten, so dass permanent ein Einklemmfall erkannt und das Schließen der Fensterscheibe blockiert werden kann. Zur Überprüfung diesbezüglich relevanter Parameter ist es bei elektrischen Antrieben im Rahmen der Entwicklung üblich, parallel zur Betätigung Informationen zu internen Berechnungen im Steuergerät auf einem CAN – Bus auszugeben. Diese CAN – Datenaufzeichnung ist neben der eigentlichen Fensterheberbetätigung ein wesentlicher Bestandteil dieses Prüfmoduls.

Die aktuelle Position einer Fensterscheibe oder Ausfälle des Antriebs am Fensterhebersystem haben automatisierungstechnisch keinen Einfluss auf die Ausführbarkeit der übrigen Prüfmodule. Fehlfunktionen führen daher nicht zur Unterbrechung der Erprobung. Eine ausführliche Dokumentation der Statusabfragen inklusive der Messungen bieten umfangreiche Möglichkeiten zur Analyse eines Fehlers und seiner Entstehung.

Der interne Ablauf dieses Prüfmoduls ist größtenteils auf andere Prüfungen mit ausschließlich elektronischer Ansteuerung, wie *Rollo*-, *Außenspiegel*- oder *Panoramadachbetätigung*, übertragbar.

#### 5.2.4 Diagnosefunktion

Das Prüfmodul *Diagnosefunktion* bietet die Möglichkeit, im Steuergerät gespeicherte Daten auszulesen oder Diagnosejobs zu starten. Die auszulesenden Daten bestehen hauptsächlich aus Fehlerspeichereinträgen, welche bei der Selbstüberwachung des Steuergeräts oder bei Störungen während der Ausführung von Diagnosejobs generiert werden. Gründe können Kabelbrüche zur angeschlossenen Sensorik oder Aktorik, das Fehlen von Informationen anderer Steuergeräte durch deren Ausfall sowie das Über- oder Unterschreiten von hinterlegten Grenzwerten sein.

Während des Lebenszyklus eines Fahrzeugs ändert sich mit jedem Fahrtantritt der Betriebsmodus der Steuergeräte von *Ruhezustand* über *Zündung* in *Fahrbetrieb*. Beim Abstellen ist die Reihenfolge umgekehrt. Einige Funktionen, beispielsweise eine Neudefinition von Grenzwerten auf Basis aufgezeichneter Messdaten, werden ereignisgesteuert nach einem Wechsel des Betriebsmodus ausgeführt, um die Alterung der Fahrzeugkomponenten bei Berechnungen im Steuergerät zu berücksichtigen. Über Diagnosejobs ist während der Dauererprobung die Möglichkeit vorhanden, Wechsel des Betriebsmodus durchzuführen und damit die Realitätsnähe der Prüfung zu steigern.

Mit Ausnahme einer Sperrung von parallel laufenden Prüfmodulen im LabVIEW – basierten Programmteil wird der überwiegende Teil der Ausführung dieses Prüfmoduls durch CANoe umgesetzt. Dazu wird für jedes involvierte Steuergerät die dazugehörige ODX – Datei benötigt, die sich mit wenigen Schritten zur *Diagnose – Konfiguration* in CANoe hinzufügen lässt. Die anschließende Zusammenstellung konkreter Abfragen und Diagnosejobs sowie die Speicherung der Ergebnisse muss durch Softwareexperten im CAPL – Programmcode erzeugt werden. Der Programmcode wird ereignisgesteuert ausgeführt, sobald durch die Ausführung des Prüfmoduls entsprechende Umgebungsvariablen in CANoe modifiziert werden.

### 5.3 Aufbau, Inbetriebnahme und Durchführung

Als Grundlage für termingerechte und aussagekräftige Erprobungsergebnisse dient ein seriennaher Aufbau, was im Besonderen die Qualität der Einzelkomponenten des Prüflings sowie deren fachgerechte Montage betrifft. Nach dem mechanischen Aufbau erfolgt die elektrische Inbetriebnahme, dabei wird die Funktionalität der relevanten Aktorik sowie Sensorik überprüft. Hier eingeschlossen ist die Kontrolle, ob alle benötigten Datenbotschaften auf den Bussystemen im Fahrzeug zur Verfügung stehen. Gegebenenfalls sind fehlende Steuergeräte mittels einer Restbussimulation einzubinden. Analog zu den Hardwareständen sind die Softwareversionen auf Aktualität zu prüfen und bei Bedarf durch aktuellere Versionen zu ersetzen. Für einen reibungslosen Ablauf der Erprobung müssen sämtliche Schutzfunktionen gegen Überbelastung, beispielsweise der Temperaturschutz bei elektrischen Antrieben, deaktiviert werden. Im Gegenzug ist eine Überwachung der Temperatur mit Abschaltfunktion oder eine Kühlung der Antriebe vorzusehen.



Steht ein erprobungsfähig aufgebauter Prüfling zu Verfügung, beginnen der Aufbau und die Konfiguration der Prüfeinrichtung. Wesentliche Bestandteile sind die Adaption der Roboterarme samt Parametrierung der Roboterbewegung (Teachen) sowie die Installation der Messhardware. Parallel dazu werden Eingangsmessungen durchgeführt, auf deren Basis später die Vergabe von Parametern für den Prüfablauf erfolgt. Die minimal benötigte Energie zum Schließen einer Tür liefert die Grundlage zur Berechnung der Schließgeschwindigkeiten.



**Abbildung 88: Aufbau einer Prüfung zur Türerprobung**

Zum Abschluss einer Inbetriebnahme gehören die Einstellungen am Prüfstandsrechner. Als erstes werden im Auswertungs- und Ansteuerungskonfigurator die am Prüfaufbau vorhandenen Sensoren und Aktoren ausgewählt, im Anschluss erfolgt die Einstellung der Parameter in der Parameterverwaltung. Als nützliches Hilfsmittel kann hierfür auf das Mess- und Analysetool zurückgegriffen werden. Der letzte Schritt ist die Zusammenstellung des Dauerlaufs im Prüfplanneditor durch die bereits implementierten Prüfmodule. Da Abläufe von typischen Prüfvorschriften häufig bereits gespeichert sind, müssen diese lediglich geladen werden.

Während der Erprobung sind durch den betreuenden Versuchsingenieur mehrmals täglich optische und akustische Kontrollen durchzuführen. Weitere Erkenntnisse über die Qualität können aus den Messdaten und Kurven mit charakteristischen Werten gewonnen werden. Eine besondere Bedeutung bekommt die Überwachung der elektrischen Antriebe, gegebenenfalls mit einer Anpassung der Pausenzeiten. Fehler bei der automatischen Versuchsdurchführung haben unmittelbar eine Benachrichtigung via SMS zur Folge. Eine permanente Überwachung der Anlage ist daher nicht nötig. Anhand der prognostizierten Laufzeiten und der Statistik in der Prüfablaufverwaltung sind Umbaumaßnahmen und Klimakammerbelegungen frühzeitig planbar. Bei gravierenden Ausfällen ist abzuwägen, ob eine Fortsetzung der Erprobung noch relevante Erkenntnisse erwarten lässt. Im Bedarfsfall sind betroffene Prüfmodule mit wenigen Klicks aus dem Prüfablauf entfernbar. Die Ereignisanzeige speichert sämtliche Informationen, die zu einem späteren Zeitpunkt für die Erstellung des Versuchsberichts relevant sind.

## 6 Realisierung eines Prüfstands für Steifigkeitsmessungen

Steifigkeitsmessungen an einem Bauteil haben zum Ziel, die Eigenschaften des verwendeten Werkstoffs und der Konstruktion anhand von Vorgaben zu überprüfen. Dafür wird an der Oberfläche des Prüflings punktuell über einen linearen Servozylinder eine kontinuierlich steigende Kraft eingeleitet und zusammen mit dem zurückgelegten Weg des Prüfstempels gemessen. Der Quotient aus Kraftänderung und Wegänderung ergibt die Steifigkeit. Mindestgrößen für die Steifigkeit an definierten Messpunkten sind im Lastenheft des Bauteils definiert.

Da an einem einzigen Bauteil, beispielsweise der Mittelkonsole im Fahrzeug, bis zu 50 verschiedene Punkte zu messen sind, stellen Roboter durch ihre Flexibilität in der Positionierung eine sinnvolle Alternative zum vergleichsweise unbeweglichen Prüfportal mit Linearantrieben dar. Zeitaufwendige Umbauarbeiten des Portals für jede einzelne Messung werden durch eine Neuausrichtung des Roboterarms und die Möglichkeit zur Speicherung von Messstellen optimiert.

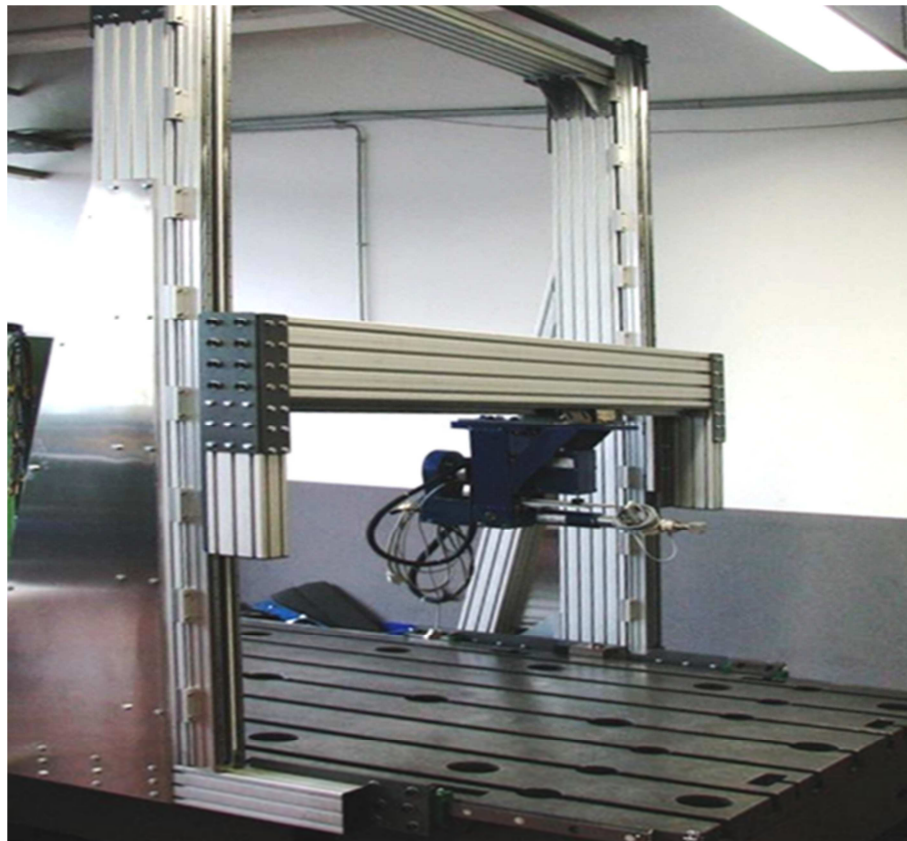


Abbildung 89: Prüfportal auf einer Messplatte mit elektrischem Servozylinder

Das Softwaretool *Steifigkeitsprüfstand* ist Bestandteil der Prüfsystemsoftware *DaLaBert* und in die softwarebasierte Systemarchitektur integriert. Somit ist der Zugriff auf sämtliche darin verfügbaren Daten und Funktionen möglich. Die Synchronisation des Roboterkoordinatensystems mit dem Fahrzeugkoordinatensystem des Prüflings gelingt mit Hilfe der Steuerbotschaft *Koordinatensystem festlegen* und ermöglicht anschließend die Verwendung der Botschaft *Position anfahren*. Dabei wird der Prüfstempel samt Roboterarm automatisch in die Ausgangsposition der anstehenden Messung positioniert. Speziell für dieses Softwaretool ist eine Funktion zur Kompensation von Messfehlern entwickelt worden, die aus einer nicht vernachlässigbaren Federsteifigkeit des Roboterarms resultieren.

## 6.1 Prüfstandsspezifische Hardware

Zur Durchführung von Steifigkeitsmessungen eignet sich besonders der Roboterarm *Stäubli RX160*. Dieser ist weitestgehend baugleich mit dem Modell *RX160L* und wird ebenfalls durch einen Robotercontroller vom Typ *CS8C* gesteuert. Im Vergleich zum *RX160L* sind aufgrund des um 300 Millimeter verkürzten Unterarms bei gleichen maximalen Kraftmomenten der Achsen höhere Kräfte am Prüfstempel erreichbar.

	
M	RX160 [Stä-11a]
R	1710mm
L/E	34kg / 248kg
W4	$\pm 270^\circ$
M1-3	692Nm, 574Nm, 242Nm

**Tabelle 38: Ausgewählter Roboterarm für Steifigkeitsmessungen**

Legende: M: Modell, R: Reichweite, L: Traglast, E: Eigengewicht, W4: Maximaler Drehwinkel der Achse 4, M1-3: Maximale Kraftmomente der Achsen 1 – 3.

Eine allgemeine Angabe zu maximal zulässigen Prüfkraften im Rahmen von Steifigkeitsmessungen ist nicht möglich. Ausschlaggebend dafür sind hauptsächlich die Kraftmomente der Achsen 1 bis 3, die sich aus der Kraft am Prüfstempel mit ihrer Wirkrichtung sowie den Hebelverhältnissen des Roboterarms ergeben. Zur Vermeidung von Überlastungen und daraus resultierenden Beschädigungen des Roboterarms ist eine Überwachung durch die Steuerbotschaft *Achsmomentexport* des Kommunikationsprotokolls implementiert.

Die Befestigung des Roboterarms erfordert eine stabile Messplatte, auf der die Prüfteile mit Hilfe von Profilschienen fixiert werden. Der Roboterarm ist mit einer Schutzhülle ausgestattet, die unter Berücksichtigung der Anforderungen aus Abschnitt 4.5.2.4 gefertigt ist. Zwischen dem Roboterflansch und dem Prüfstempel ist ein piezoelektrisches 3 – Komponenten – Kraftmesselement vom Typ *9327C* der Firma *Kistler* eingebaut. Der Sensor verfügt über einen Messbereich von 8.000 Newton in Stoßrichtung und ist in der Lage, durch Querkkräfte verursachte Momente von 22 Newtonmetern unbeschadet zu überstehen [Kis-10b]. Dies ist notwendig, da kein anderweitiger mechanischer Schutz für den Sensor existiert. Abbildung 90 zeigt den Prüfstand bei einer Steifigkeitsmessung sowie den dazugehörigen Prüfstandsrechner.





Abbildung 90: Prüfstand für Steifigkeitsmessungen

## 6.2 Kompensation der Federsteifigkeit des Roboterarms

Vergleiche zwischen der Wegerfassung des roboterinternen Messsystems und externen Sensoren ergaben kraftabhängige Abweichungen von mehreren Millimetern. Ursächlich dafür ist die Federrate des Roboterarms, welche im Wesentlichen aus einer Biegung im Ober- und Unterarm sowie dem Getriebespiel resultiert. Abbildung 91 zeigt den Aufbau der Vergleichsmessung, der externe Sensor ist hier nicht sichtbar hinter der senkrecht eingespannten Metallplatte positioniert.



Abbildung 91: Versuchsaufbau zum Vergleich der Wegmessung zwischen roboterinterner und externer Sensorik

Zur Kompensation der Federsteifigkeit wird die Wegdifferenz  $W_d$  zwischen dem roboterseitig gemessenen Weg  $W_R$  und der tatsächlich zurückgelegten Strecke  $W_e$  auf Basis von Vergleichsmessungen berechnet. Der Ansatz zur Korrektur der Daten des roboterinternen Messsystems besteht darin, die berechnete Wegdifferenz  $W_d$  von den während der Prüfung gemessenen Werten  $W_R$  zu subtrahieren und so den tatsächlich zurückgelegten Weg  $W_{R_k}$  zu bestimmen.

Die Wegdifferenz  $W_d$  ist primär von der Höhe der eingeleiteten Kraft  $F$ , der Richtung der Roboterbewegung  $\vec{r}$  sowie dem Abstand  $d$  vom Kraftangriffspunkt zum Ursprung des Roboterkoordinatensystems abhängig. Abbildung 92 vergleicht den kraftabhängigen Verlauf der Wegmessung des Roboters  $W_R(F)$  mit der tatsächlich zurückgelegten Strecke  $W_e(F)$ . Der Richtungsvektor  $\vec{r}$  zeigt dabei in X – Richtung des Roboterkoordinatensystems, die Messung erfolgt im Abstand von 940 Millimetern zum Ursprung.

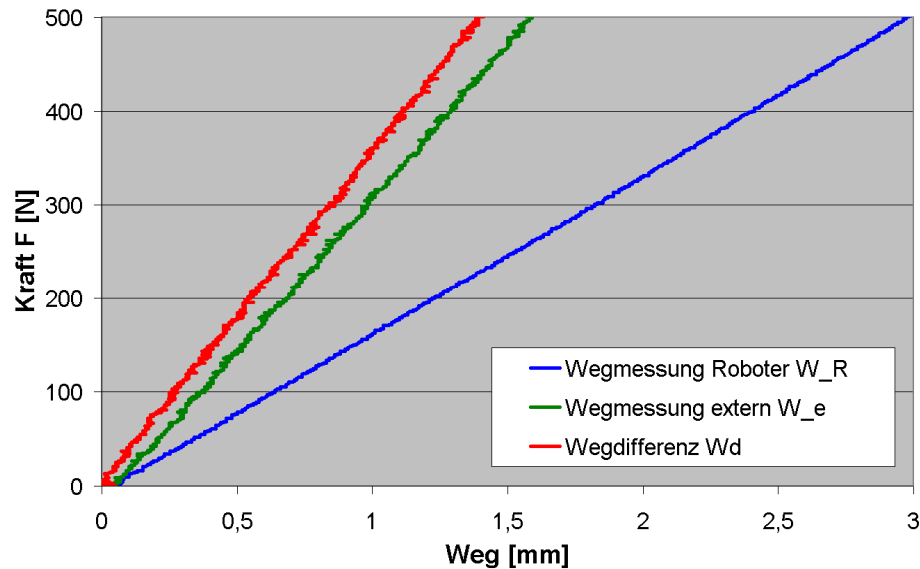


Abbildung 92: Vergleich der Wegmessung zwischen roboterinterner und externer Sensorik

Abbildung 92 rechtfertigt die Annahme, dass zwischen der Kraft  $F$  und der Wegdifferenz  $W_d$  ein linearer Zusammenhang besteht. Zur Bestätigung dieser Erkenntnis wurden in X-, Y- und Z – Richtung im Roboterkoordinatensystem jeweils acht Vergleichsmessungen im Abstand von 100 Millimetern durchgeführt. Dabei konnte dieser lineare Zusammenhang durchgehend bestätigt werden. Die Auswertung der Messkurven ergab weiterhin, dass die Steigung der Wegdifferenz  $W_d$  eine Funktion des Abstands  $d$  der Krafteinleitung zum Ursprung des Roboterkoordinatensystems sowie der dazugehörigen Richtung  $\vec{r}$  ist. Abbildung 93 zeigt den Verlauf der Steigung bei einer Krafteinleitung in X – Richtung.

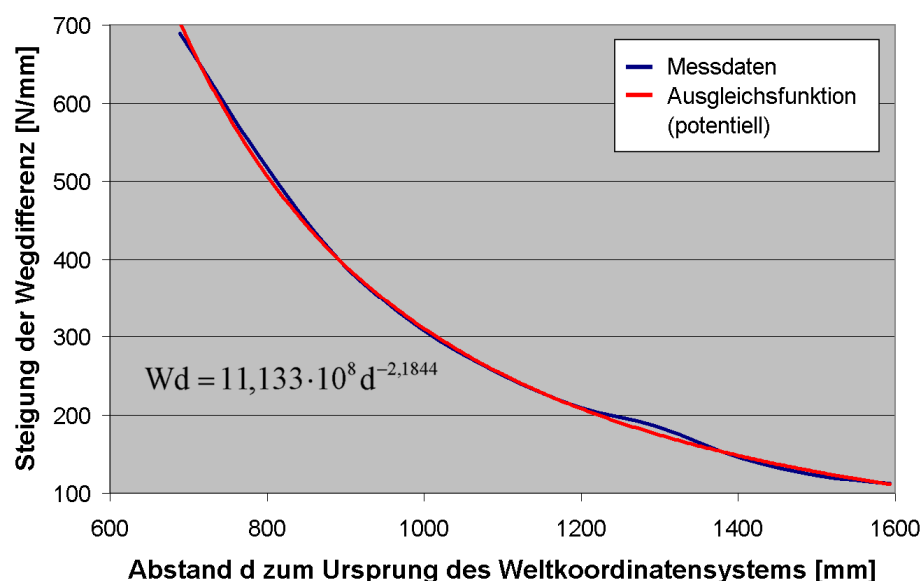


Abbildung 93: Steigung der Wegdifferenz in Abhängigkeit des Abstands in X – Richtung

Vergleiche zwischen einer linearen, logarithmischen, polynomischen, exponentiellen und potentiellen Ausgleichsfunktion zeigen, dass letztgenannte sehr genau dem Verlauf der experimentell ermittelten Daten entspricht. Abbildung 93 zeigt die potentielle Ausgleichsfunktion mit dazugehöriger Gleichung. Eine Approximation der Wegdifferenz  $W_d$  in Abhängigkeit von der Kraft  $F$  und dem Abstand  $d$  durch die folgende Formel liefert hinreichend genaue Ergebnisse.

$$W_d(F, d) = \frac{F}{a \cdot d^B}$$

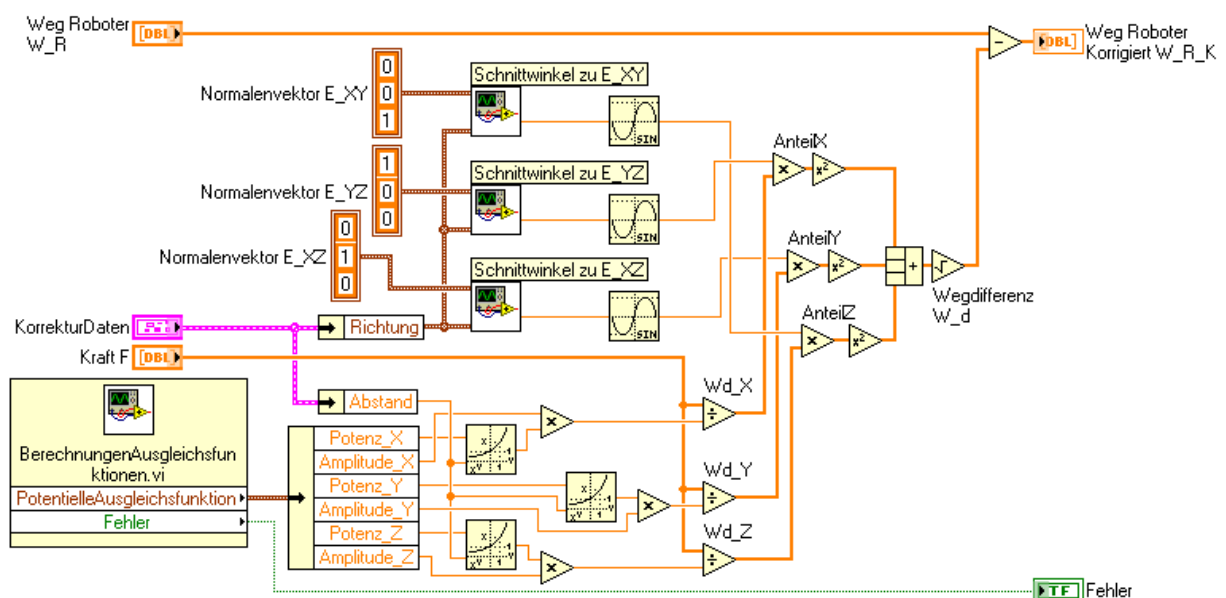
**Formel 11: Funktion der Wegdifferenz**

Die Berechnung der Amplitude  $a$  sowie der Potenz  $B$  basiert auf den bereits beschriebenen 24 Vergleichsmessungen, deren repräsentativen Verlauf Abbildung 92 dargestellt. Zur Minimierung von Einflüssen durch Messungenauigkeiten kommt zuerst die in LabVIEW verfügbare Funktion zur Bestimmung einer linearen Ausgleichsgerade mit jeweils zehn Wertepaaren für Kraft  $F$  und Wegdifferenz  $W_d$  im Bereich von 0 bis 500 Newton zum Einsatz. Zur Kontrolle der Berechnungsergebnisse dient die Verschiebungskonstante der linearen Funktion. Diese muss annähernd null ergeben, da aus einer Kraft  $F$  von 0 Newton noch keine Wegdifferenz  $W_d$  hervorgehen darf. Mit den so erhaltenen abstandsabhängigen Steigungen einer Richtung  $\vec{r}$  sind die Berechnungen von Amplitude  $a$  und Potenz  $B$  der potentiellen Ausgleichsfunktion möglich. LabVIEW stellt dafür ebenfalls eine entsprechende Funktion nach der Methode *kleinste Fehlerquadrate* bereit.

Mit Hilfe der Schnittwinkel  $\alpha_{XY}$ ,  $\alpha_{XZ}$  und  $\alpha_{YZ}$  des Richtungsvektors  $\vec{r}$  mit den drei Ebenen  $E_{XY}$ ,  $E_{XZ}$  und  $E_{YZ}$  des Roboterkoordinatensystems lassen sich die Einflüsse der richtungsabhängigen Wegdifferenzen  $W_{d_X}$ ,  $W_{d_Y}$  und  $W_{d_Z}$  zur daraus resultierenden Wegdifferenz  $W_d$  aufsummieren.

$$W_d = \sqrt{(W_{d_X} \cdot \sin(\alpha_{YZ}))^2 + (W_{d_Y} \cdot \sin(\alpha_{XZ}))^2 + (W_{d_Z} \cdot \sin(\alpha_{XY}))^2}$$

**Formel 12: Richtungsabhängige Wegdifferenz**



**Abbildung 94: Programmcode zur Kompensation der Federsteifigkeit**



Abbildung 94 beschreibt den Programmcode zur Berechnung der Wegdifferenz. Das Unterprogramm *BerechnungAusgleichsfunktionen.vi* generiert mit Hilfe der Daten von Vergleichsmessungen die Amplituden und Potenzen der potentiellen Ausgleichsfunktionen in allen drei Raumachsen. Durch Anwendung der Formel 11 erfolgt anschließend die Berechnung der richtungsabhängigen Wegdifferenzen. Die jeweiligen Anteile an der gesamten Wegdifferenz werden im nächsten Schritt analog zur Formel 12 summiert und dann vom gemessenen Weg des Roboters subtrahiert. Fehler beim Aufrufen der Vergleichsmessungen führen zum Funktionsabbruch.

Diese Berechnungsmethode kompensiert die Abweichung zwischen dem vom Robotersystem gemessenen und dem realen Weg bei einer Belastung von 0 bis 500 Newton für Zug- und Druckversuche. Vergleichsmessungen im Rahmen eines Ringversuchs zwischen dem roboterbasierten Steifigkeitsprüfstand mit Kompensationsrechnung und zwei weiteren Prüfständen haben maximale Abweichungen der Steifigkeit von zwei Prozent ergeben. Eine Abweichung dieser Größenordnung wird unter Berücksichtigung der gängigen Prüfvorschriften als akzeptabel betrachtet. Der roboterbasierte Steifigkeitsprüfstand ist als Erprobungssystem freigegeben.

Eine alternative Kompensationsmethode würde die Bestimmung der Federrate des Roboterarms darstellen, um diese nach der Messung mit der Steifigkeit des zu prüfenden Bauteils zu verrechnen. Die Ermittlung der Federsteifigkeit des Roboterarms wäre durch einen gesonderten Steifigkeitsprüfstand oder durch eine Vergleichsmessung mit separater Wegaufnahme an einer Feder mit bekannter Konstante möglich. Beide Varianten sind in der Praxis mit wesentlich mehr Aufwand verbunden, als die geschilderte Berechnung der Wegdifferenz durch Vergleichsmessungen. Hinzu kommt, dass die Prüfberichte grundsätzliche Kraft – Weg – Diagramme enthalten und daher eine direkte Korrektur der Eingangssignale am sinnvollsten ist.

## 6.3 Bedienoberfläche

Das Softwaretool *Steifigkeitsprüfstand* wird in einem separaten Fenster ausgeführt und beinhaltet vier Reiter. Der Reiter *Kompensation Wegfehler* umfasst die Einstellung der externen Wegsensoren sowie die Auswahl der zur Kompensationsrechnung benötigten acht Messpunkte für jede Richtung. Zwei Graphen zeigen direkt im Abschluss an eine Messung die Kurven von Kraft und Wegen und ermöglichen dadurch eine sofortige Kontrolle. Im Bedarfsfall ist eine Wiederholung der Messungen in beliebiger Reihenfolge möglich.

Verschleißbedingte Veränderungen der Federsteifigkeit des Roboterarms sind aus diesen Messdaten nicht ersichtlich, da kein Vergleich mit vormals aufgezeichneten Messkurven erfolgt. Eine derartige Überprüfung obliegt den Softwareexperten, vorherige Vergleichsmessungen werden aus diesem Grund vor dem Überschreiben archiviert. Mit Hilfe archivierter Messungen einer Position ist im Laufe der Zeit eine Aussage über die zu erwartenden Änderungen der Federsteifigkeit möglich. Ebenso wird deutlich, unter welchen Umständen eine Erneuerung der Kompensationsdaten notwendig ist.

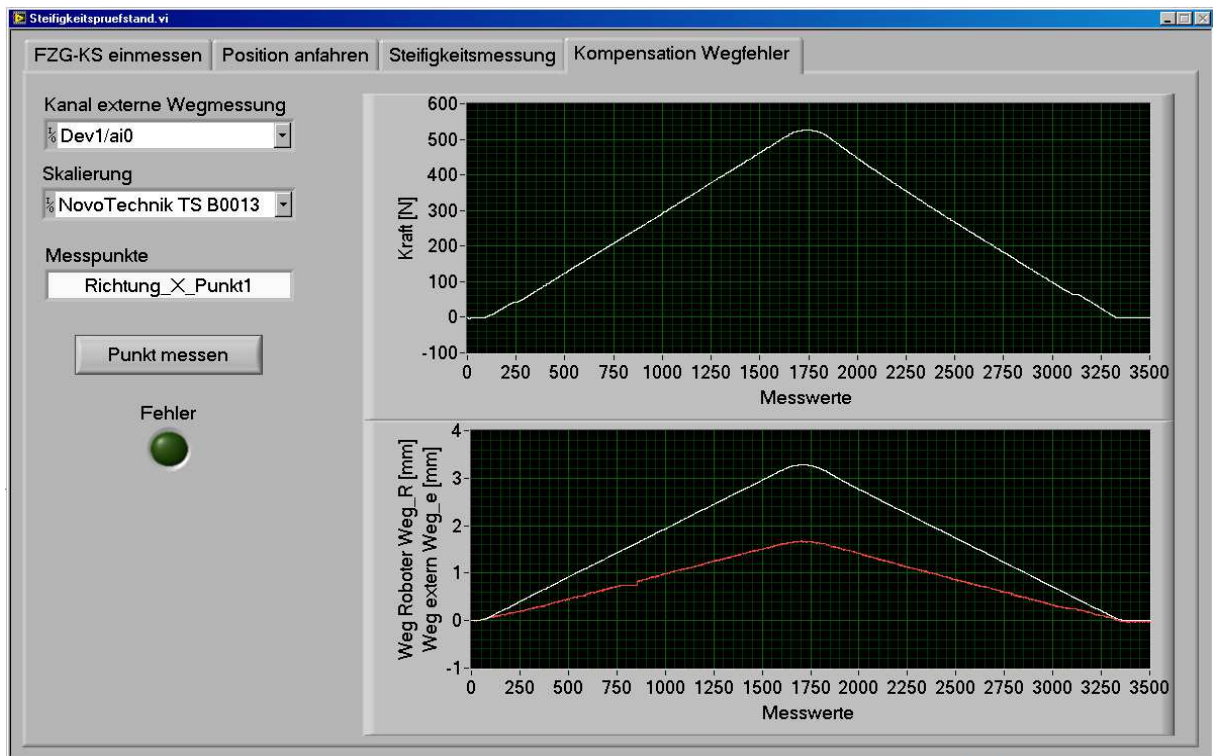


Abbildung 95: Bedienoberfläche zur Ausführung der Vergleichsmessungen zur Kompensation der Federsteifigkeit des Roboterarms

Abbildung 96 zeigt den Reiter zur Positionierung des Roboterarms an einer durch das Lastenheft definierten Position im Fahrzeugkoordinatensystem. Die Zielposition kann manuell durch die Bedienelemente *Punkt* und *Richtung* eingestellt werden, bei mehreren Messpunkten empfiehlt sich der Listenimport über eine Datei. Das benötigte Format ist auf der Bedienoberfläche abgedruckt. Der Parameter *DeltaZ* gibt den Offset zur Zielposition in Stoßrichtung an, damit beim Start der Messung keine Berührung zwischen Prüfstempel und Prüfling vorliegt.



Abbildung 96: Bedienoberfläche zur Positionierung des Prüfstempels

Ein automatisches Anfahren der Messpunkte wäre prinzipiell umsetzbar und somit die Voraussetzung zur vollständigen Automatisierung von Steifigkeitsmessungen. Aktuell ist diese Erweiterung jedoch nicht vorgesehen. Hierzu würden umfangreiche Sicherheitsmaßnahmen benötigt, um eine Beschädigung der Prüfteile beim automatischen Anfahren einer Position zu verhindern. Die Überwachung einer Prüfung durch den Versuchsingenieur und der dadurch mögliche manuelle Eingriff bieten einen besseren Schutz der Prototypenteile und Einzelanfertigungen. Prüfstandsbedingte Beschädigungen an Bauteilen, die nur in sehr begrenzter Stückzahl zur Verfügung stehen, würden hohe Zusatzkosten und im schlimmsten Fall Verzögerungen im Freigabeprozess bedeuten.

Die eigentliche Durchführung der Messung mit anschließender Steifigkeitsberechnung erfolgt im gleichnamigen Reiter. Dafür sind die zum Prüfpunkt definierten Werte für Geschwindigkeit und Kraft einzutragen. Der Parameter *Strecke* ist als Schutz konzipiert und sollte maximal 50 Prozent oberhalb des zu erwartenden Wegs liegen.

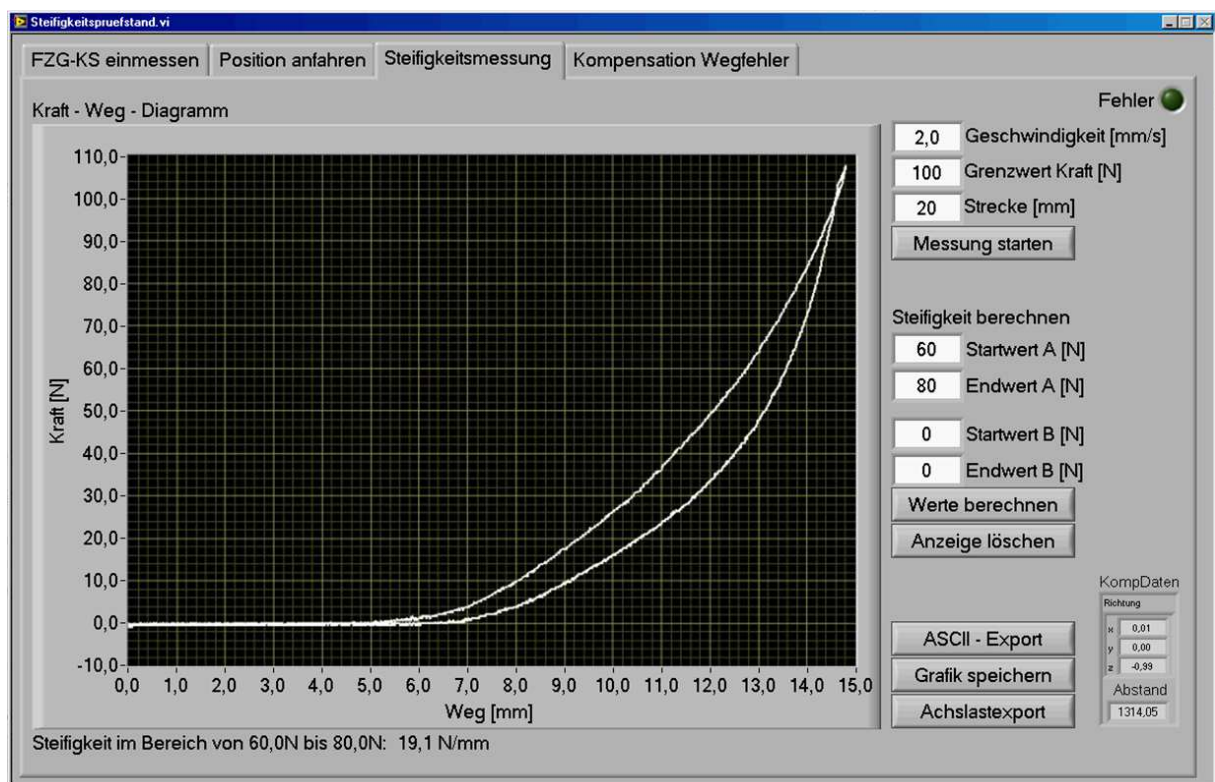


Abbildung 97: Bedienoberfläche zur Ausführung einer Steifigkeitsmessung

Im Anschluss an die Messung kann die Steifigkeit des Bauteils am ausgewählten Punkt für zwei verschiedene Kraftbereiche A und B berechnet werden. Die Ausgabe erfolgt unterhalb der Messkurve. Ein Export der angezeigten Werte im ASCII – Format zur Weiterverarbeitung oder als Grafik zur direkten Übernahme in den Prüfbericht wird angeboten. Zur Überwachung der Kraftmomente an den Achsen des Roboterarms ist ein Export der parallel zur Messung über die Steuerfunktion *Achsmomentexport* empfangenen Daten möglich.

## 7 Bewertung

Der Erfolg eines Produktes ist im Allgemeinen von drei Bedingungen anhängig: Als erste Voraussetzung sind die Anforderungen des Verwenders zu erfüllen, zweitens muss das Produkt rechtzeitig zur Verfügung stehen und drittens hat der Preis gerechtfertigt zu sein [Pah-07].

Sämtliche in Abschnitt 4.1 aufgeführten *Forderungen* und *Wünsche* werden durch die im Verlauf dieser Arbeit beschriebenen Lösungen erfüllt. Das Prüfsystem steht bereits seit Beginn des Jahres 2011 in der Praxis zur Verfügung und wird seitdem permanent eingesetzt und erweitert. In Abschnitt 7.3 sind die Investitionskosten dargestellt, anhand derer unter Berücksichtigung der individuellen Rahmenbedingungen eine Amortisationsrechnung möglich ist.

### 7.1 Prüfsystemkonzept

Die inhaltlichen Schwerpunkte des im Rahmen dieser Arbeit entworfenen, entwickelten und validierten Prüfsystemkonzepts sind die Softwarearchitektur, die Verwaltung von Messdaten, die aufgabenorientierte Schnittstelle zur Integration der Robotik sowie die Hardwarekomponenten zum Aufbau von konkreten Prüfständen.

Die Softwarearchitektur hat sich als flexibel, zweckmäßig und laufstabil im praktischen Einsatz bewährt. Insbesondere beim Hinzufügen oder Erweitern von Prüfmodulen zeigt sich ihr Reifegrad, da keine strukturellen Änderungen erforderlich waren. Beispiele sind die nachträgliche Erweiterung zur CAN – Datenaufzeichnung im Prüfmodul *Fensterheberbetätigung* oder die Erweiterung des Prüfmoduls *Türbewegung* um das Türkonzept der außergewöhnlichen Variante des *Mini Clubman* aus dem Jahr 2007. Bei diesen und vergleichbaren Änderungsanforderungen bestätigt sich die Qualität der Softwarearchitektur durch einen geringen Zeitaufwand zur Implementierung der neuen Funktionen. In diesem Zusammenhang hat sich LabVIEW als äußerst effektiv erwiesen. Aufgrund der grafischen Programmieroberfläche in Kombination mit dem modularen Aufbau ist eine schnell verständliche Lesbarkeit des Programmcodes gegeben. Dies erleichtert Ergänzungen und vermindert grobe Programmierfehler. Die Laufstabilität der verwendeten Softwareprogramme ist essenziell bei Lebensdauerversuchen. Die Ausführung von LabVIEW und CANoe auf einem Rechner mit Microsoft Windows – Minimalinstallation hat diesbezüglich sowie unter Berücksichtigung der weichen Echtzeit keine Schwächen aufgezeigt. Während des gesamten Einsatzes kam es zu keinen nennenswerten Programmabstürzen und somit zu keinerlei Verzögerungen.

Die ausgearbeiteten Methoden zur Aufzeichnung, Verwaltung und Analyse von Messdaten tragen wesentlich dazu bei, ein Höchstmaß an Erkenntnissen aus den Lebensdauerprüfungen zu gewinnen. Bauteildefekte können nun konkreten Belastungszyklen zugeordnet werden, die jeweils vorherrschenden Umweltbedingungen sind protokolliert. Aus der jetzt vorhandenen Menge an Messdaten lassen sich Tendenzen zur erwartenden Lebensdauer ableiten. Die lückenlose Protokollierung aller Ereignisse ist die Grundlage für eine Reproduzierbarkeit des Prüfablaufs und dient als Nachweis für eine der Prüfvorschrift entsprechenden Versuchsdurchführung. Über die grafische Aufbereitung und prägnante Präsentation der aussagekräftigsten Messkurven ist eine ständige und effektive Versuchsüberwachung durch die Ver-

suchsingenieure möglich. Manuelle Vergleichsmessungen relevanter interner und externer Signale unterstützen in optimaler Weise die Fehlersuche und Parametrierung von Prüfmodulen.

Die aufgabenorientierte Schnittstelle zur Integration der Robotik erweitert die Anwendungsmöglichkeiten von Robotern im Versuchswesen. Damit eröffnet sich auch Mitarbeitern ohne umfangreiche Roboterkenntnis und -erfahrung die Möglichkeit, einfachste Bewegungsabläufe selbständig in Betrieb zu nehmen. Die aufgabenorientierte Schnittstelle sowie die darauf basierenden Bedienoberflächen sind der optimale Mittelweg zwischen einer großen Flexibilität in der Anwendung und einer möglichst hohen Abstraktion für die Bedienung. Bei komplexen Bewegungsabläufen, erstmalig ausgeführten Prüfungen und zusätzlich erforderlichen Sensoren werden auch zukünftig Roboter- und Softwareexperten Unterstützung leisten müssen. Nur unter diesen Bedingungen ist der Einsatz von Robotik im Versuchswesen langfristig profitabel. Die Schnittstelle zur Robotik wurde im Rahmen dieser Arbeit an Robotersystemen des Herstellers *Stäubli* mit *CS8C* – Controller entwickelt und getestet, da dieser Controllertyp aktuell über den leistungsfähigsten Rechner und einen entsprechenden Funktionsumfang in der herstellerspezifischen Programmiersprache verfügt. Zukünftig ist bei weiteren Roboterherstellern mit ähnlich leistungsstarken Controllern zu rechnen, was den Einsatz alternativer Robotertypen ermöglicht.

Der Einsatz von Robotik zur mechanischen Betätigung trägt wesentlich zur Durchführbarkeit von flexiblen Bewegungsabläufen und zur Effizienzsteigerung des Prüfsystems bei. Die Rüstzeiten haben sich signifikant von mehreren Wochen auf wenige Tage verkürzt. Anpassungen an den jeweiligen Fahrzeugtyp sind lediglich bei der Adaption der Roboter im Fahrzeuginnenraum notwendig. Roboter sind erprobte Standardkomponenten, daher gehören prüfsystembedingte Ausfälle und langwieriges Fehlersuchen an den Spezialkonstruktionen der Vergangenheit an. Extreme Umweltbedingungen in der Klimakammer verursachen bei den Messmitteln, Roboterwerkzeugen und Schutzhüllen starke Beanspruchungen, bei diesen Komponenten müssen frühzeitige Ausfälle einkalkuliert werden. Zur Vermeidung prüfstandsbedingter Verzögerungen sind daher Ersatzteile vorzuhalten. Reparaturen der Roboter sind zeitaufwendig und kostenintensiv, aus diesem Grund sollte mindestens nach jedem Test eine Sichtprüfung auf Beschädigungen oder Abnutzungserscheinungen erfolgen.

Die im Rahmen dieser Arbeit entwickelten Strukturen und Methoden können neben dem Anwendungsgebiet der Fahrzeugentwicklung auch in anderen Bereichen zu Optimierungen beitragen. Die softwarebasierte Systemarchitektur ist erwiesenermaßen eine ideale Grundlage für Automatisierungssysteme, die eine schnelle und flexible Inbetriebnahme von Prozessen mit experimentellem Hintergrund und umfangreichen Möglichkeiten zur Fehleranalyse bei vielfältigen Ein- und Ausgangssignalen bereitstellen müssen. Der beschriebene Abstraktionsgrad der Bedienoberfläche dient als Vorlage, um ein bestimmtes Maß an Flexibilität sicherzustellen, falls die Bediener über keinerlei Programmierkenntnisse verfügen. Die Methodik im Umgang mit Messdaten ist sinnvoll für Bereiche, in denen die Aufnahme, Bewertung und Überwachung von nicht vorhersehbaren Messgrößen und Verläufen mit einer hohen Abhängigkeit von externen Einflussgrößen erfolgen. Dies gilt prinzipiell für alle Arten von Lebensdaueruntersuchungen, aber ebenso für die Überwachung von Produktionsprozessen, falls die Anwendung statistischer Berechnungen nicht umsetzbar ist und trotzdem Aussagen hinsichtlich einer Tendenz zu treffen sind. Die aufgabenorientierte Schnittstelle zur Robotik ist grundsätzlich unabhängig vom Einsatzgebiet. Sie bringt besonders dort Vorteile, wo die Inbetriebnahme einer Handhabungs-, Bearbeitungs- oder Messaufgabe aufgrund der Genauigkeit, Schnelligkeit und Ausdauer eines Roboters durch Bediener mit geringen Kenntnissen in der Roboterprogrammierung oder im Umgang mit Koordinatensystemen einzustellen ist. Die dazu entwickelte grafische Bedienoberfläche dient als Basis, die abhängig von der konkreten Prob-



ematik angepasst werden muss. Einsatzbeispiele sind unterstützende Tätigkeiten bei der Produktion von Kleinserien oder in Umweltlaboren.

## 7.2 Einsatz in der Praxis

Zwischen Dezember 2007 und Juni 2012 wurden 35 Dauerversuche zur Ermittlung der Langzeitbetriebsfestigkeit an Fahrzeugtüren durchgeführt. Die Fahrzeugmodelle erstrecken sich dabei vom Kleinwagen bis zur Luxuslimousine verschiedener Fabrikate. Der Nachweis einer Praxistauglichkeit ist damit erbracht.



Abbildung 98: Bilderserie durchgeführter Türdauerläufe

Die Einzelbilder in Abbildung 98 entstanden über den gesamten Entwicklungszeitraum, sie zeigen verschiedene Baustände der Roboterschutzhüllen, Roboterwerkzeuge und Fahrzeugadaption sowie Roboter der Hersteller *Stäubli* und *Mitsubishi*.



Im gleichen Zeitraum wurden weitere Einzelprüfungen von Fensterhebersystemen, Außenspiegeln, Schiebedächern, Front- und Heckklappen mit Hilfe der Software *DaLaBert* erfolgreich durchgeführt.

Über den Entwicklungszeitraum hat sich die Effizienz des Prüfsystems schrittweise bis hin zur optimalen Leistungsfähigkeit verbessert. Zu Beginn war für einen Prüfzyklus, bestehend aus den Prüfmodulen *Türbewegung*, *Fensterheberbetätigung* und *Türschlossbetätigung*, ein Zeitaufwand von 45 Sekunden einzuplanen. Eine typische Türerprobung inklusive Aufbau, Inbetriebnahme und Analyse benötigte einen Zeitraum von etwa zwei Monaten. *Kinderkrankheiten* in der Programmierung trugen zu dieser Durchlaufzeit bei. Heute findet der beschriebene Prüfzyklus in 28 Sekunden statt, prüfsystembedingte Ausfälle sind sehr selten und zeitlich zu vernachlässigen. Die zweckmäßige Adaption der Prüfstandshardware, die standardisierte Schnittstelle zur Fahrzeugelektronik sowie die gesammelten Erfahrungen der Versuchsmitarbeiter ermöglichen im Optimalfall eine Durchlaufzeit von drei Wochen. Besonders die Funktion der SMS – Benachrichtigung von Mitarbeitern im Störfall hat massiv zur Minimierung der Prüfungsunterbrechungen beigetragen. Die Forderung nach einer zeitgerafften Erprobung wird damit gegenüber den zuvor eingesetzten Prüfständen in einem wesentlich höheren Maße erfüllt.

Mit Blick auf die Realitätsnähe und Aussagekraft der Erprobung ist zu berücksichtigen, dass bestimmte Standzeiten bei verschiedenen Umweltbedingungen in der Klimakammer zu erbringen sind. Bei sehr kurzen Zykluszeiten sind notwendigerweise Pausen in den Prüfablauf einzufügen, um Belastungen durch Temperatureinflüsse oder Korrosion abzusichern und mit vormaligen Erprobungsschleifen vergleichen zu können.

Das Mess- und Analysetool hat sich zur Unterstützung der Inbetriebnahme, für eine Parameteroptimierung sowie für kurzfristig durchzuführende Messungen im Fehlerfall als leistungstarkes Hilfsmittel innerhalb des Prüfsystems etabliert. Ebenso gilt dies für die Untersuchung von prüfsysteminternen Fehlfunktionen, die durch stetige Anpassungen und Erweiterungen nicht vollständig auszuschließen sind. Des Weiteren hat sich die Ausweitung zur kontinuierlichen Messdatenaufzeichnung mit anschließender Archivierung bewährt. Schadteilanalysen oder Untersuchungen von Fehlfunktionen durch Hersteller oder Zulieferfirmen im Anschluss an die Erprobungen konnten so bereits durch präzise Aussagen unterstützt werden.

Der Einsatz von CANoe setzt voraus, dass zum Fahrzeugmodell passende Datenbanken und die Restbussimulation vom Hersteller vorliegen. Dies ist in der Praxis problemlos umgesetzt, die benötigten Daten stehen kurzfristig und in funktionsfähiger Form zur Verfügung. Der Kompromiss, die von CANoe übernommenen Funktionalitäten nicht eigenständig und damit unabhängig zu programmieren, hat sich als ideal erwiesen.

Die intuitive Bedienbarkeit, die vielfältigen Schnittstellen, das Mess- und Analysetool sowie die hohe Laufstabilität des Prüfsystems auf Standardrechnern führten schnell zur großen Verbreitung im Versuchswesen. Der Einsatz erfolgt auch bei Prüfungen, für welche die Software *DaLaBert* auf den ersten Blick überdimensioniert erscheint. Beispielhaft ist die Ansteuerung einer Wasserpumpe zur Prüfung einer Scheibenreinigungsanlage. Diese Automatisierung besteht lediglich aus der Beschaltung eines Relais und einer anschließenden Pause. Dieser Umstand zeigt die hohe Akzeptanz der Versuchsmitarbeiter für das neue Konzept. Neben den beschriebenen Prüfständen für Karosseriekomponenten und Steifigkeitsmessungen ist mittlerweile ebenfalls ein Sitzprüfstand im Einsatz.

Die Verfügbarkeit zusätzlicher Prüfvarianten durch das neue Prüfsystemkonzept führte bereits zu Optimierungen an bestehenden Prüfvorschriften. Bei bestimmten Türmodellen sind inzwi-

schen verschiedene Kraftangriffspunkte bei der Betätigung vorgeschrieben. Die Flexibilität einer Roboterbewegung macht die Umsetzung problemlos möglich. Zudem ist die Absicherung der Türinnenverkleidung in die Absicherung des Türrohbaus integriert. Dies wurde möglich, da mit Hilfe der roboterbasierten Türinnenbetätigung keine untypische Modifikation der Türinnenverkleidung mehr notwendig ist.

## 7.3 Investitionsbetrachtung

In der Automobilindustrie ist es üblich, einmalige Prüfungen, Betreibermodelle zur Durchführung von Erprobungsaufträgen über längere Zeiträume oder auch komplette Fahrzeugentwicklungsaufträge an Fremdfirmen zu vergeben. Als Voraussetzung für die Vergabe derartiger Leistungsumfänge muss der Ingenieurdienstleister über die benötigten technischen Anlagen verfügen. Im Vorfeld einer Investition in kostenintensive Prüfanlagen ist daher abzuwägen, ob durch die kalkulierten Einnahmen in einem überschaubaren Zeitraum eine Amortisation zu erwarten ist.

Die Materialkosten für einen Prüfstand zur Erprobung von Fahrzeugtüren belaufen sich auf insgesamt 380.000 Euro. Tabelle 39 beschreibt die wesentlichen Einzelpositionen.

Beschreibung	Preis
Robotik 4 Roboter zur Türaußenbetätigung <i>Stäubli RX160L</i> 2 Roboter zur Türinnenbetätigung <i>Stäubli TX60L</i>	240.000€
Fahrzeugadaption der Robotik 4 Podeste, 2 Profilschienen für Komplettfahrzeuge 1 Grundgestell zur Aufnahme von Lackiertraversen	12.000€
6 Roboterschutzhüllen mit Belüftungssystem	80.000€
Roboterwerkzeuge 4 Werkzeuge zur Außenbetätigung, 2 Werkzeuge zur Innenbetätigung	3.000€
Mobile Mehrfach – Sicherheits – Lichtschranken <i>Sick M4000</i>	7.000€
Prüfstandsrechner Schaltschrank, Industrierechner, Messkarten, Netzwerkkomponenten Batterieladecomputer, Anschlusstechnik, Elektronikbedarf	8.000€
Messtechnik 6 Stromsensoren, 4 Drehwinkelgeber, 4 Näherungssensoren 6 Kraftsensoren mit Ladungsverstärker	11.000€
Software LabVIEW, CANoe mit Hardwareschnittstelle <i>VN8900</i>	16.000€
Sonstiges Verbrauchsmaterial	3.000€
<b>Materialgesamtkosten</b>	<b>380.000€</b>

Tabelle 39: Materialkostenaufstellung

Die Kostenaufstellung aus Tabelle 39 verdeutlicht, dass ein Großteil der Materialkosten auf die Robotik entfällt. Dadurch verringert sich das Risiko der Anschaffung, da Roboter auch anderweitig einsetzbar sind. Nicht berücksichtigt sind die Anschaffungskosten für eine Klimakammer, da diese ebenso für anderweitige Prüfungen genutzt werden kann und üblicherweise nicht zum Türprüfstand zählt.

Ein gleichfalls erheblicher Kostenpunkt sind die Arbeitsstunden zur Konstruktion und Fertigung der Roboterwerkzeuge und Fahrzeugadaption sowie der Aufwand für Programmierung, Test und Optimierung der Prüfsystemautomatisierung. Im Rahmen dieses Promotionsprojekts wurden drei Mannjahre benötigt, bis das Gesamtsystem in einer praxistauglichen Version zur Verfügung stand.

## 8 Zusammenfassung und Ausblick

Mit dem in dieser Arbeit entwickelten Prüfsystemkonzept für das Versuchswesen der Automobilentwicklung ist eine realitätsnähere sowie zeit- und kostensparendere Durchführung von realen Versuchen zur Erprobung von Karosserieelementen möglich. Das neuartige Konzept ist eine Reaktion auf die Anforderung, Innovationen in der Automobiltechnik bedarfsgerecht abzusichern. Die Verkürzung der Entwicklungszeit und die daraus resultierende zeitliche Überlagerung von Konstruktion, Berechnung und Realversuch sowie eine Zunahme elektronischer Funktionen im Automobil erforderten eine grundlegende Überarbeitung der Prüfmethoden.

Kern des Prüfsystemkonzepts ist eine in LabVIEW umgesetzte softwarebasierte Systemarchitektur mit Schwerpunkten in den Bereichen Anwendungsflexibilität, Analysefunktionalität, Erweiterbarkeit und intuitive Bedienung. Konkrete Prüfungen sind softwaretechnisch als Prüfmodule programmiert und können zu einem kombinierten und dadurch realitätsnahen Dauerlauf zusammengestellt werden. Das Prüfsystemkonzept ist in der Lage, umfangreiche Anforderungen verschiedenster Prüfvorschriften zu erfüllen. Die Devise für den Bediener lautet *Konfigurieren statt Programmieren*, welche zu einer Verringerung von Fehlerquellen bei der Bedienung führt. Die Verwendung der Standardsoftware CANoe macht das Prüfsystem unabhängig von herstellerspezifischer Fahrzeugelektronik, ein neues Fahrzeugmodell ist in wenigen Stunden elektronisch adaptiert. Durch eine umfangreiche Messdatenaufzeichnung kann das Prüfsystem automatisch auf Veränderungen am Prüfling reagieren. Zudem verhindert die Überwachung relevanter Kenngrößen besonders bei elektrischen Bauteilen eine Überbelastung. Optimierungen hinsichtlich des kundennahen Prüfverhaltens und das Eliminieren von untypischen Belastungen haben eine Reduktion von prüfstandsbedingten Schäden am Prüfteil zur Folge. Die Dokumentationsfunktion mit SMS – Benachrichtigung hat eine signifikante Verringerung der ungeplanten Prüfungsunterbrechungen bei Dauerläufen erbracht. Aus dieser Fülle an Verbesserungen und Überwachungsfunktionen resultieren Prüfungsunterbrechungen von durchschnittlich weniger als zehn Prozent sowie eine deutliche Verringerung des Betreuungsaufwands.

Der Einsatz von Robotern zur Ausführung mechanischer Betätigungen verleiht eine hohe Flexibilität. Lange Rüstzeiten und teure Spezialvorrichtungen gehören der Vergangenheit an. Zur Einbindung der Roboter in den Prüfablauf dient eine auf das Versuchswesen abgestimmte aufgabenorientierte Schnittstelle. Diese ermöglicht Mitarbeitern ohne tiefgreifende Robotikerfahrungen, in zwei aufgabenabhängigen Varianten eine *einfachste Betätigung* oder einen *komplexen Bewegungsablauf* einzustellen. Die Realisierung zweier Roboterwerkzeuge sowie eine Vorrichtung zur Adaption der Roboter am Prüfling ermöglichen die mechanische Betätigung von Fahrzeugtüren. Eine mitentwickelte Roboterschutzhülle sichert dabei den Betrieb unter extremen Klimabedingungen. Der roboterbasierte Steifigkeitsprüfstand zeigt Ansätze zur Effektivitätssteigerung derartiger Messungen durch eine Erhöhung des Automatisierungsgrads. Zudem verbessert sich die Vergleichbarkeit mit Ergebnissen aus der Berechnung.

Das Projekt wurde in enger Kooperation mit verantwortlichen Fachstellen zur Fahrzeugerprobung bei einem Automobilhersteller durchgeführt. Die Prüfeinrichtung unterlag einer fortlaufenden Erweiterung und Optimierung während einer Vielzahl von praktischen Einsätzen bei Lebensdauererprobungen und Einzelmessungen.

Spannend bleibt die Frage, wie lange noch Realversuche Bestandteil der Fahrzeugentwicklung sein werden, bevor die Ablösung durch eine vollständig virtuelle Erprobung stattfindet.

Aus heutiger Sicht sprechen noch einige Argumente dagegen. Dazu gehören endliche Rechnerkapazität, begrenzte Modelldetaillierung sowie unbekannte physikalische Zusammenhänge bei Simulationsberechnungen. Ebenso sind gesetzliche Bestimmungen einzuhalten. Während der Mitarbeit im Versuchswesen zeigten sich teilweise gravierende Unterschiede zwischen dem Sollzustand einer Konstruktion und dem real gefertigten Prototyp. Zur Absicherung neuartiger Konzepte, Werkstoffe und Fertigungsverfahren sowie zur Optimierung von Simulationsmodellen werden auf lange Sicht Realversuche notwendig sein.

Die Automobilindustrie lebt von Innovationen, zudem führen unerwartete Ereignisse im Verlauf einer Prüfung regelmäßig zu Abweichungen vom ursprünglich geplanten Vorgehen. Das Prüfsystem muss daher fortwährend auf Basis der zugrunde liegenden Architektur an neue Anforderungen angepasst werden. Dadurch ist zukünftig ein effektiver und rentabler Einsatz gesichert. Die Flexibilität der Roboter ermöglicht theoretisch eine Verknüpfung verschiedenster Teilprüfungen, beispielsweise das Drücken von Bedienelementen im Innenraum oder Verstellungen am Lenkrad, mit einer Türerprobung. Praktisch ist zu berücksichtigen, dass jedes zusätzliche Prüfmodul die Zykluszeit sowie die Gefahr von Ausfällen im Prüfablauf erhöht.

In der Praxis verwendete Klimakammern sind für eine Simulation von Temperatur und Luftfeuchte ausgelegt. Denkbar ist die Erzeugung weiterer Wetterverhältnisse, wie Sonnenlicht, Wind, Regen oder Schnee. Überlagerte Vibrationsprüfungen würden eine weitere Belastungsart integrieren und Erprobungszeiten einsparen. Mit Blick auf die zuverlässige Abarbeitung eines Erprobungsauftrags sind jedoch extreme Umweltbedingungen erfahrungsgemäß die größte Herausforderung.

Aufgrund der Reduzierung von Erprobungsschleifen bei zukünftigen Fahrzeugprojekten wird es zunehmend wichtiger, aus einem Versuch ein Maximum an Erkenntnissen zu gewinnen. Eine Grundlage hierfür ist die umfassende und kontinuierliche Messdatenerfassung. Sofern es die Umweltbedingungen zulassen, könnten auch optische und akustische Aufzeichnungen in den Prüfablauf integriert werden. Grundsätzlich ist zu beachten, dass eine automatisierte Messdatenerfassung bei Langzeitversuchen binnen kurzer Zeit zu einer unüberschaubaren Datenmenge führen kann. Für diesen Fall sind Methoden und Funktionen zur praxisgerechten Datenanalyse erforderlich. Ein konkretes Konzept ist in dieser Arbeit beschrieben. Die größte Gefahr einer maximalen Automatisierung von Lebensdauertests besteht darin, dass sich Versuchingenieure zu sehr auf die Auswertungen des Prüfsystems verlassen. Bei der Erprobung von Prototypen können jederzeit unvorhersehbare Ereignisse auftreten. Zur Beurteilung von Prüflingen und für eine Entscheidung über das weitere Vorgehen bei Störungen ist ein hohes Maß an Fachwissen und Erfahrung notwendig. Dies ist nicht durch automatisierte Prüfsysteme zu leisten.

Eine Konsequenz der ausgeweiteten Messdatenerfassung und -analyse ist eine zweckmäßige Aufbereitung und Archivierung zur späteren Verwendung. Fehlfunktionen, die bei hohen Erprobungszyklen auftreten, sind im Feld erst nach einigen Produktionsjahren zu erwarten. Die Verantwortung zur Fehlerbehebung liegt dann nicht mehr bei der Entwicklung, sondern im Produktionswerk und nach Ende der Produktionsdauer bei den Werkstätten. Um die Arbeit der Werkstätten zu erleichtern und letztendlich Unzufriedenheit vom Kunden abzuwenden, ist zukünftig ein verbesserter Wissenstransfer zwischen den jeweils verantwortlichen Abteilungen zielführend, um die immer komplizierter erscheinenden Fehlerbilder zu beheben.

Mittelfristig ist geplant, das Prüfsystemkonzept als niederlassungsweites Standard – Werkzeug für Erprobungsaufträge in der Karosserie- und Interieurerprobung einzusetzen. Eine dahin gehende Weiterentwicklung hat bereits begonnen, ebenso wie die Ausweitung des Ein-

satzgebietes auf Niederlassungen innerhalb der Bertrandt – Gruppe zur Absicherung von Komponenten weiterer namhafter Fahrzeughersteller sowie deren Systemlieferanten. Eine Voraussetzung dafür ist die Verfügbarkeit von Softwareexperten zur Betreuung der Prüfsysteme am Einsatzort. Eine zentrale Verwaltung bedeutet eine zu große räumliche Entfernung vom konkreten Problem und steht dadurch effektiven und kurzfristigen Anpassungen entgegen. Die Nähe zwischen Softwareexperte und Versuchingenieur ist ein wesentlicher Erfolgsfaktor.



## 9 Thesen

1. Innovationen in der Fahrzeugtechnik erfordern eine stetige Anpassung der Prüfsystemkonzepte zur Gewährleistung einer angemessenen realitätsnahen Absicherung. Hauptursachen sind die Zunahme von Elektronikkomponenten, die steigende Anzahl der Derivate sowie die immer kürzeren Entwicklungszeiten. Die Gründe dafür liegen in den Marktbedingungen und wachsenden Kundenforderungen an qualitativ hochwertige, komfortable, sichere sowie umweltfreundliche Automobile.
2. Der Anteil der virtuellen Erprobung in der Fahrzeugentwicklung steigt stetig, dennoch wird der Realversuch auf lange Sicht unverzichtbar sein. Beide Verfahren haben ihre Legitimation und verfügen über Vorteile, aber auch Grenzen in den verschiedensten Phasen eines Entwicklungsprozesses. Eine Symbiose der jeweiligen Stärken ist die Grundlage für eine optimale und fortschrittliche Erprobung.
3. Parallel zu wenigen Standardprüfständen existieren in der Komponentenerprobung viele Prüfstände als *Einzellösung*. Diese verursachen hohe Anschaffungskosten bei geringer anderweitiger Verwendbarkeit und Auslastung. Als Basis für verschiedenste Prüfungsaufgaben ist ein allgemeines auf das Versuchswesen zugeschnittenes Prüfsystemkonzept notwendig.
4. Bei Prüfsystemen mit hohen Flexibilitäts- und lediglich *weichen* Echtzeitanforderungen ist eine PC – basierte Konfiguration einem SPS – basierten Aufbau vorzuziehen. Softwarebasierte Lösungen sind in der Regel schneller veränderbar, komfortabler bedienbar und weniger kostenintensiv. PC – Systeme besitzen vielfältige Hard- und Softwareschnittstellen und Vorteile bei der Datenverarbeitung, Visualisierung und Vernetzung.
5. Die Softwarearchitektur mit der zentral im Hauptspeicher installierten *Datenbasis* ermöglicht jederzeit eine direkte Aufnahme und Überwachung sämtlicher Zustandsgrößen im Prüfsystem. Die Datenflüsse sowie die Ermittlung des Prüflingsstatus durch Eingangsgrößen verschiedenster Datentypen sind so genau nachvollziehbar. Dies gewährleistet eine äußerst effektive Analyse von Fehlfunktionen bei automatisierten Abläufen mit experimentellem Charakter. Dieses Konzept unterstützt die einfache Erweiterung der Ein- und Ausgangsgrößen.
6. Ein modular und zweckmäßig strukturierter Aufbau der softwarebasierten Systemarchitektur ist die Grundlage für eine hohe Zuverlässigkeit im Gebrauch sowie die kurzfristige Implementierung neuer Funktionen. Dadurch sind eine zeitgeraffte Lebensdauererprobung und die kurzfristige Reaktion auf die verschiedensten Fahrzeug- und Ausstattungsvarianten sowie auf unvorhersehbare Ereignisse gewährleistet.

7. Eine Abstraktionsschicht zwischen den Hard- und Softwareschnittstellen und dem *Prüfplaneditor*, realisiert durch den *Auswertungs-* und den *Ansteuerungskonfigurator*, verleiht dem Prüfsystem eine hohe Flexibilität. Notwendige situationsbedingte Änderungen bei der Auswertung oder Ansteuerung des Prüflings lassen sich unmittelbar vornehmen.
8. Die einheitliche und zentrale Speicherung sämtlicher Parameter des Prüfsystems in der *Datenbasis* sichert den schnellstmöglichen Zugriff sowie die Möglichkeit zur automatisierten Änderung der Werte. Dadurch kann theoretisch jeder Parameter manuell oder im Zusammenhang mit Steuer- und Regelprozessen mit sofortiger Wirkung online modifiziert werden.
9. Die *Prüfmodule* beinhalten den Programmcode zur Ausführung einer konkreten Aktion auf Basis der Prüfvorschriften. Die Entwicklung durch Softwareexperten ist individuell und enthält die Ansteuerung und Überwachung des Bauteils, die Messdatenerfassung sowie die Auswertung. *Prüfmodule* bestehen soweit wie möglich aus vorgefertigten *Basisprogrammen*. Jedes zusätzliche Prüfmodul steigert die Leistungsfähigkeit des Prüfsystems, eine stetige Erweiterung durch die beständige Veränderung der Rahmenbedingungen ist konzeptionell vorgesehen.
10. Eine Ausweitung der Messdatenerfassung trägt dazu bei, Versuchsergebnisse zu objektivieren und die Überwachung sowie Reproduzierbarkeit von Prüfabläufen zu verbessern. Die daraus resultierende große Ansammlung von Messdaten erfordert jedoch zweckmäßige Methoden zur Auswertung, Analyse und Weiterverarbeitung. Eine frei konfigurierbare *Messdatenpräsentation* ermöglicht es den Versuchingenieuren, in wenigen Minuten die Beschaffenheit des Prüflings anhand der maßgeblichen Messkurven unter Berücksichtigung der Umgebungsbedingungen zu kontrollieren.
11. Der Automatisierungsgrad von Prüfsystemen für Prototypentests ist so zu wählen, dass in letzter Instanz qualifizierte Mitarbeiter eine Bewertung vornehmen und kritische Entscheidungen treffen. Blindes Vertrauen in vollständig automatisierte Prüfabläufe, Messungen und darauf aufbauenden Analysefunktionen führt zu unerkannten Fehlern. Bei freigaberelevanten Absicherungen von Fahrzeugkomponenten ist dies keinesfalls vertretbar.
12. Die Informationsdienste mit SMS – Benachrichtigung im Fehlerfall minimieren fehlerbedingte Prüfungsunterbrechungen signifikant. Statistisch ermittelte Terminvorhersagen verbessern die Planung von bevorstehenden Messungen, Analysen oder Umbaumaßnahmen am Prüfaufbau. Beide Maßnahmen führen zu einer erheblichen Verkürzung der Durchlaufzeit einer Prüfung.

13. Roboter sind grundsätzlich sehr nützliche Hilfsmittel im Versuchswesen, wobei die Art und Weise der Programmierung wesentlich über den Erfolg entscheidet. Eine aufgabenorientierte Schnittstelle unter Einbeziehung der externen Sensorik sowie eine intuitive grafische Oberfläche ist die Lösung, Robotertechnik auch durch Mitarbeiter ohne tiefgreifende Erfahrung effektiv einzusetzen. Für *komplexe Bewegungsabläufe*, beispielsweise dem Öffnen und Schließen einer Fahrzeugtür, bedarf es eines *Grundgerüsts*, welches die bedienerseitig einzulernenden Punkte und Parameter definiert.
14. Die Durchführung von Steifigkeitsmessungen mit Hilfe von Robotern ist prinzipiell möglich, allerdings muss die Federrate des Roboterarms berücksichtigt werden. Daraus resultierende Abweichung zwischen dem real zurückgelegten und dem vom Robotersystem gemessenen Weg kann mit Hilfe eines mathematischen Modells in wesentlicher Abhängigkeit der Bewegungsrichtung, der Armstellung und der wirkenden Kraft auf Basis von Vergleichsmessungen berechnet werden.
15. Prüfsysteme müssen ein optimales Verhältnis zwischen der Flexibilität im Einsatz und dem Abstraktionsgrad für die Bedienung aufweisen. In der Entwicklungsphase ist daher eine intensive Abstimmung zwischen dem Versuchsingenieur und dem Softwareexperten unabdingbar und maßgeblich für den wirtschaftlichen Erfolg verantwortlich.
16. Bereits nach kurzer Einsatzzeit sind die hohen Belastungen durch extreme klimatische Bedingungen an den Hardwarekomponenten eines Prüfstands zur Lebensdauererprobung sichtbar. Bauteildefekte und Ausfälle müssen stets einkalkuliert werden. Die konsequente Verwendung von Standardkomponenten in allen Bereichen, auch bei Software, spart Kosten und minimiert Ausfallzeiten.

## 10 Quellenverzeichnis

- [Abb-11] Produktdatenblatt ABB IRB 2600 Industrieroboter. Friedberg: ABB Automation GmbH, 2011
- [Abe-02] Abele, R.: Die Windeseile auf dem Kutschbock. Frankfurter Allgemeine Sonntagszeitung, 19.05.2002. Frankfurt: F.A.Z. Verlag, 2002  
Die Originalquelle lag nicht vor.
- [Ade-10] Produktdatenblatt Adept Viper s1700. Dortmund: Adept Technology GmbH, 2010
- [Adg-09] ADG Automatisierung Dresden GmbH: Steuerung verteilter paralleler Prüfstände mit PXI und LabVIEW – RT. Konzept und Erfahrungen. 7. Technologietag „Prüfstandskonzepte in der Automobilindustrie“. National Instruments, 2009
- [Afo-06] Afonso, G.; Pires, N.; Estrela, N.: Force control experiments for industrial applications: a test case using an industrial deburring example. Technical School of Viseu (Portugal), 2006
- [Ahl-04] Ahlbehrendt, N. et al.: Sensortechnik für den ortsflexiblen Robotereinsatz. Karlsruhe: Projektträger Karlsruhe 2004
- [Ahr-09] Ahrends, S.: Innovative Werkzeuge für die Mess- und Prüftechnik. 7. Technologietag „Prüfstandskonzepte in der Automobilindustrie“. National Instruments, 2009
- [Alm-07] Almeida, C.: Programmierbares 3D – Werkstückerkennungssystem für die sensorgeführte Roboterhandhabung. Dissertation, Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 2007
- [Ans-01] Anselm, D.: Die PKW – Karosserie. Konstruktion, Deformationsverhalten, Unfallinstandsetzung. 2. Auflage, Würzburg: Vogel Verlag, 2001
- [Arm-06a] Armbruster, H.; Kirner, E.; Kinkel, S.: Neue Nutzungspotentiale für Industrieroboter – Ergebnisse einer Betriebsbefragung. wt Werkstattstechnik online 96, Heft 9. Düsseldorf: Springer VDI Verlag, 2006
- [Arm-06b] Armbruster, H.; Kirner, E.; Kinkel, S.: Neue Kundengruppen für Industrieroboter. PI Mitteilung Nr.: 38. Karlsruhe: Fraunhofer Institut für System- und Innovationsforschung, 2006
- [Ata-08] Atamer, S.; Barenbrock, D.; Ageorges, C.; Sonsino, C.: Berücksichtigung der Resonanzeffekte bei der Betriebsfestigkeitsauslegung von PKW – Rohbaukarosserien. ATZ Automobiltechnische Zeitschrift 110, Heft 9. Wiesbaden: Vieweg Verlag, 2008

- [Aut-09] Automation 2009. Der Automatisierungskongress in Deutschland. Tagung in Baden – Baden. VDI Berichte Nr. 2067. Düsseldorf: VDI Verlag, 2009
- [Aut-11] Internetseite des Automobile Club de l'Ouest (ACO), 24 Stunden von Le Mans: <http://www.lemans.org/en/races/24h/history.html>. Abgerufen am 10.11.2010
- [Avl-10] Internetseite der Firma AVL List GmbH: <https://www.avl.com/company> Abgerufen am 15.10.2010
- [Bac-08] Bacher, R.; Haas, H.; Simons, M.: Integration von FlexRay – basierenden Steuergeräten in vorhandene Prüfstände. ATZelektronik 3, Heft 1. Wiesbaden: Vieweg Verlag, 2008
- [Bah-08] Bahner, J.: Übersteigertes Vertrauen in Automation: Der Einfluss von Fehlererfahrungen auf Complacency und Automation Bias, Dissertation, Berlin: Technische Universität Berlin, 2008
- [Bal-00] Balzert, H.: Lehrbuch der Software – Technik. Band 1: Software-Entwicklung. 2. Auflage, Heidelberg: Spektrum Akademischer Verlag, 2000
- [Bar-01] Bartels, J.; Vogel, B.: Systementwicklung für die Automatisierung im Anlagenbau. . AT Automatisierungstechnik 49, Heft 5. München: Oldenbourg Wissenschaftsverlag, 2001
- [Bar-11] Bartols, F.: Restbussimulation von Time – Triggered Ethernet in Automotive Anwendungen: Verwandte Arbeiten. Hamburg: Hochschule für Angewandte Wissenschaften Hamburg, 2011
- [Bat-05] Battenberg G.; Kreipl, C.: Werttreiber Messrobotik. Querschnittstechnologie mit hohem Diffusionspotential. ZfAW 8, Heft 1. Bamberg: FAW Verlag, 2005
- [Bat-06] Bathelt, J.: Entwicklungsmethodik für SPS – gesteuerte mechatronische Systeme. Dissertation, Zürich: Eidgenössische Technische Hochschule Zürich, 2006
- [Bat-08] Battenberg, G.: Millionenfach berührt. Prüfrobotik für harte Ansprüche in der Landmaschinentechnik. Automation und Qualität 6 / 2008. Hannover: Schlütersche Verlagsgesellschaft, 2008
- [Bat-10] Battenberg: Produktdatenblatt: Erster Klima-Prüfschrank mit integrierter Messrobotik WK BM 1000. Marburg: Battenberg ROBOTIC GmbH & Co. KG, 2010
- [Bau-09] Bauer, C.; Frings, K.; Harsche, J.: Unternehmen in schwierigen Zeiten – Strategien des hessischen Mittelstands. Hessischer Mittelstandsbericht 2009. Wiesbaden: HA Hessen Agentur GmbH, 2009
- [Bei-06] Beimert, H. P.: Testen und Simulieren von Baugruppen. ATZelektronik 1, Heft 3. Wiesbaden: Vieweg Verlag, 2006
- [Ben-86] Benz & Co. in Mannheim: Patentschrift Nr. 37435 (Kaiserliches Patentamt). Carl Benz Motorwagen Nr. 1, ausgegeben am 02.12.1886

- [Ben-11a] Bender, M.: Verfahren zur Industrieroboterprogrammierung – Onlineprogrammierverfahren. Siegen: Universität Siegen, 2011
- [Ben-11b] Benz Patent Motorwagen Nummer 1: [http://de.wikipedia.org/wiki/Benz\\_Patent-Motorwagen\\_Nummer\\_1](http://de.wikipedia.org/wiki/Benz_Patent-Motorwagen_Nummer_1). Abgerufen am 31.05.2011
- [Ber-08] Berechnung und Simulation im Fahrzeugbau. Tagung in Baden – Baden. VDI Berichte Nr. 2031. Düsseldorf: VDI Wissensforum GmbH, 2008
- [Ber-09] Bertsche, B.; Göhner, P.; Jensen, U.; Schinköthe, W.; Wunderlich, H.: Zuverlässigkeit mechatronischer Systeme. Grundlagen und Bewertung in frühen Entwicklungsphasen. Berlin: Springer Verlag, 2009
- [Bet-09] Bethscheider – Kieser, U.: Simulation – unverzichtbar in der Automobilindustrie. Automotive IT, 05.2009. Hannover: Media – Manufaktur Verlag, 2009
- [Beu-04] Beu, A.: Intuitive und attraktive Bedienoberflächen für Maschinen. atp Automatisierungstechnische Praxis 46, Heft 9. München: Oldenbourg Industrieverlag, 2004
- [Beu-05] Beumelburg, K.: Fähigkeitsorientierte Montageplanung in der direkten Mensch – Roboter – Kooperation. Dissertation, Stuttgart: Universität Stuttgart, 2005
- [Bic-00] Bickendorf, J.: Neue Anwendungsgebiete für Industrieroboter durch CAD – basierte Offline – Programmierung. VDI Berichte Nr. 1552: Robotik, Düsseldorf: VDI Verlag, 2000
- [Bin-09] Bindel, T.; Hofmann, D.: Projektierung von Automatisierungsanlagen. Wiesbaden: Vieweg Verlag, 2009
- [Bis-07] Bischoff, R.; Kogan, Y.; Kurth, J.: Lebenszyklusorientierte Betrachtung der Einsatzmöglichkeiten von Augmented Reality in der Industrierobotik. atp Automatisierungstechnische Praxis 49, Heft 7. München: Oldenbourg Industrieverlag, 2007
- [Blo-10] Blochowicz, T.; Laeri, F.: Einführung in die Messdatenanalyse für das physikalische Grundpraktikum. Darmstadt: Technische Universität Darmstadt, 2010
- [Bmw-03] Virtuelle Realität im Automobilbau. Information der BMW Group. München, 2003
- [Bmw-11] Bilder aus der Produktion des BMW 1er, Karosseriebau. Regensburg: BMW Werk, 2011
- [Boc-04] Bock, T.: Von stationären Robotern in der Vorfertigung zu humanoiden Robotern auf der Baustelle. atp Automatisierungstechnische Praxis 46, Heft 12. München: Oldenbourg Industrieverlag, 2004



- [Boc-08] Bock, T.; Maurer, M.; van Meel, F.; Müller, T.: Vehicle in the Loop – Ein innovativer Ansatz zur Kopplung virtueller mit realer Erprobung. ATZ Automobiltechnische Zeitschrift 110, Heft 1. Wiesbaden: Vieweg Verlag, 2008
- [Bon-03] Bonhoff, N.; Rüsche, H.-J.: Türmodule – Autotüren schlüsselfertig. ATZ Automobiltechnische Zeitschrift 105, Heft 9. Wiesbaden: Vieweg Verlag, 2003
- [Bon-10] Bonnaventure de, R.: High – Speed Application. Schneller Steuergerätezugang am Prüfstand – Eine Lösung von ETAS und D2T. elektronik industrie 6/2010. Heidelberg, Hüthig Verlag, 2010
- [Bra-89] Braden, R.: Requirements for Internet Hosts – Communication Layers. Request for Comments: 1122. Internet Engineering Task Force, 10.1989 <http://www.ietf.org/rfc/rfc1122.txt>
- [Bra-01] Braess, H. – H.; Seiffert, U.: Handbuch Kraftfahrzeugtechnik. 2. Auflage, Wiesbaden: Vieweg Verlag, 2001
- [Bra-03] Braess, H. – H.: Das Automobil und die Wissenschaften – Von der Erkenntnis zur Realität, von der Realität zur Erkenntnis. ATZ Automobiltechnische Zeitschrift 105, Heft 3. Wiesbaden: Vieweg Verlag, 2003
- [Bra-07] Braess, H. – H.; Seiffert, U.: Automobildesign und Technik. ATZ / MTZ Fachbuch, Wiesbaden: Vieweg Verlag, 2007
- [Bre-04] Brecher, C. et al.: Ein flexibel einsetzbares Robotersystem für variierende Aufgaben in der Maschinenbeschickung. VDI Berichte Nr. 1841: Robotik, Düsseldorf: VDI Verlag, 2004
- [Bre-05] Brecher, C.; Schröter, B.: Inbetriebnahme und Programmierung portabler Robotersysteme. Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 2005
- [Bre-08a] Brecher, C.; Müller, R.: Rent – A – Robot: Vorstellung des Forschungsprojekts. Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 2008
- [Bre-08b] Brecher, C.; Müller, R.; Schuh, G.: Rent – A – Robot. Abschlussbericht des BMBF – Verbundprojekts. Aachen: Apprimus Verlag, 2008
- [Bre-09a] Brecher, C. et al.: Modellbasierte Programmierung „by Demonstration“ – Schnelle Inbetriebnahme von Robotersystemen. atp Automatisierungstechnische Praxis 51, Heft 7. München: Oldenbourg Industrie Verlag, 2009
- [Bre-09b] Brecher, C.; Göbel, M.; Herfs, W.; Pohlmann, G.: Roboterprogrammierung durch Demonstration - Ein ganzheitliches Verfahren zur intuitiven Erzeugung und Optimierung von Roboterprogrammen. wt Werkstattstechnik online 99, Heft 9. Düsseldorf: Springer VDI Verlag, 2009
- [Bre-10a] Brecher, C. et al.: Intuitive Roboterprogrammierung in der automatisierten Montage. Ein hybrides Verfahren zur Programmierung durch direkte Interaktion. wt Werkstattstechnik online 100, Heft 9. Düsseldorf: Springer VDI Verlag, 2010

- [Bre-10b] Bredendiek, J.: RS – 232 – Interface. Grundlagen: <http://www.sprut.de/electronic/interfaces/rs232/rs232.htm>. Abgerufen am 17.10.2010
- [Bre-06] Breitling, T.; Dragon, L.; Grossmann, T.: Digitale Prototypen: Ein weitere Meilenstein zur Verbesserung der Abläufe und Zusammenarbeit in der PKW – Entwicklung. VDI Berichte Nr. 1967: Berechnung und Simulation im Fahrzeugbau, Düsseldorf: VDI Verlag, 2006
- [Bri-07] Brinkmann, C.: 24 Millionen Testkilometer: Die Erprobung der neuen Mercedes C – Klasse. Ahaus: Speed Heads, 19.01.2007
- [Bru-94] Brunner, B.; Arbter, K.; Hirzinger, G.: Aufgabenorientierte Programmierung von sensorbasierten Robotersystemen. KI – 94 Workshops Saarbrücken, 18. – 23. September, 1994
- [Bru-01] Brunner, B.; Vogel, J.; Hirzinger, G.: Aufgabenorientierte Fernprogrammierung von Robotern. AT Automatisierungstechnik 49, Heft 7. München: Oldenbourg Wissenschaftsverlag, 2001
- [Buc-11] Buchholz, P.; Bause, F.: Modellgestützte Analyse und Optimierung. Script zur Vorlesung Modellbildung, Simulation und Analyse. Dortmund: Technische Universität Dortmund, 2011
- [Bur-02] Burmberger, G.: PC – basierte Systemarchitekturen für zeitkritische technische Prozesse. Dissertation, München: Technische Universität München, 2002
- [Bur-10] Burgdorf, F.: Eine kunden- und lebenszyklusorientierte Produktfamilienabsicherung für die Automobilindustrie. Dissertation, Karlsruhe: Karlsruher Institut für Technologie, 2010
- [Com-07] Produktdatenblatt COMAU Smart NS 16 – 1.65. Dreieich: COMAU Deutschland GmbH, 2007
- [Dai-09] Daimler: Der neue Mercedes – Benz SLS AMG. Konzeption und Entwicklung: Virtuell und real – digitale und echte Prototypen für höchste Ansprüche. AMG besitzt Verantwortung für Konzeption und Entwicklung. Laguna Seca (USA), 02.11.2009
- [Dam-96] Dammertz, R.: Ein Programmiersystem zur graphisch strukturierten Erstellung von Roboterprogrammen und Programmieroberflächen. Dissertation, Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 1996
- [Dam-09] Damm, B.: Roboter mit Navigationsgerät. Tools 1.2009. Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 2009
- [Dat-10] DAT – Report 2010. Deutsche Automobil Treuhand GmbH. Würzburg: Vogel Business Media, 2010
- [Dei-11] Deiterding, J.: Simplified Integration of External Sensors in Industrial Robot Programs. Dissertation, Bayreuth: Universität Bayreuth, 2011

- [Den-04] Denkena, B. et al.: Vereinfachte Programmierung von Industrierobotern. wt Werkstattstechnik 94, Heft 9. Düsseldorf: Springer VDI Verlag, 2004
- [Den-05] Denkena, B. et al.: Roboterprogrammierung in der Fertigung. wt Werkstattstechnik online 95, Heft 9. Düsseldorf: Springer VDI Verlag, 2005
- [Deu-06] Deuschl, M.: Gestaltung eines Prüffelds für die Fahrwerksentwicklung unter Berücksichtigung der virtuellen Produktentwicklung. Dissertation, München: Technische Universität München, 2006
- [Deu-09] Deutronic: Wandlungsfähige Prüfautomaten. Modularer Aufbau macht Testsysteme flexibel. Design & Elektronik 10.2009. Kissing: WEKA Media, 2009
- [Deu-10] Produktdatenblatt Ladecomputer / Intelligent Charging Computer DBL800. Adlkofen: Deutronic Elektronik GmbH, 2010
- [Dil-00] Dillmann, R.: Programmieren durch Vormachen in der Robotik. Karlsruhe: Universität Karlsruhe, 2000
- [Dil-06] Dillmann, R.: SFB 588 – Humanoide Roboter – Lernende und kooperierende multimodale Roboter. Vortrag am 17.11.2006. Karlsruhe: Universität Karlsruhe, 2006
- [Dom-97] Domschke, W.; Scholl, A.; Voß, S.: Produktionsplanung. Berlin: Springer Verlag, 1997
- [Dre-03] Dreher, A.: Ethernet in der industriellen Datenkommunikation. Anwendungen mit „harten“ Echtzeitanforderungen. atp Automatisierungstechnische Praxis 45, Heft 4. München: Oldenbourg Industrieverlag, 2003
- [Dav-11] David, D.: The Story of the Grand Prix. Internetseite: <http://www.ddavid.com/formula1/story.htm>. Abgerufen am 22.4.2011
- [Egg-10] Eggers, S.; Kuther, T.: Wie Sie Kfz – Steuergeräte in nicht vollständig zur Verfügung stehenden FlexRay – Bordnetzen testen. ELEKTRONIKPRAXIS 623. Würzburg: Vogel Business Media, 2010
- [Ehr-03] Ehrenmann, M.; Rogalla, O.; Zöllner, R.; Dillmann, R.: Belehrung komplexer Aufgaben für Serviceroboter: Programmieren durch Vormachen im Werkstätten- und Haushaltsbereich. Karlsruhe: Universität Karlsruhe, 2003
- [Ehr-06] Ehrmann, M.; Seckner, M.: Ein Ansatz zur nutzergerechten Programmierung von Industrierobotern. atp Automatisierungstechnische Praxis 48, Heft 4. München: Oldenbourg Industrieverlag, 2006
- [Erb-06] Erben, M.; Günther, W.; Seldmeier, T.; Lederer, D.; Rueß, H.: Haftung bei Fehlern in der Simulations-/ Berechnungssoftware, insbesondere aufgrund von Produkthaftung. VDI Berichte Nr. 1967: Berechnung und Simulation im Fahrzeugbau, Düsseldorf: VDI Verlag, 2006

- [Erp-05] Erprobung und Simulation in der Fahrzeugentwicklung. Kongress in Würzburg. Düsseldorf: VDI Verlag, 2005
- [Erp-07] Erprobung und Simulation in der Fahrzeugentwicklung. Kongress in Würzburg. Düsseldorf: VDI Verlag, 2007
- [Fan-10] Produktdatenblatt FANUC M-20iA Serie. Neuhausen a.d.F.: FANUC Robotics Deutschland GmbH, 2010
- [Fel-06] Felchtinger, G.: Automation auf engstem Raum. atp Automatisierungstechnische Praxis 48, Heft 8. München: Oldenbourg Industrieverlag, 2006
- [Fah-10] Fahrversuch – Süd: Fahrzeugdauerlauf – Was ist Dauerhaltbarkeits- und Funktionsvalidierung beziehungsweise Fahrzeugdauerlauf? Schwäbisch Gmünd: Fahrversuch Süd GmbH, 2010
- [Fin-05] Finkemeyer, B.; Kröger, T.; Wahl, F.: Aktionsprimitive: Ein universelles Roboter – Programmierparadigma. AT Automatisierungstechnik 53, Heft 4. München: Oldenbourg Wissenschaftsverlag, 2005
- [Fra-10] Fraunhofer IPA: Industrieroboteranwendungen Montage: Trends in der Handhabungstechnik. Stuttgart: Fraunhofer – Institut für Produktionstechnik und Automatisierung, 2010
- [Fre-90] Freund, E.; Heck, H.; Kreft, K.; Mauve, C.: OSIRIS – Ein objektorientiertes System zur impliziten Roboterprogrammierung. Robotersysteme Band 6. Berlin: Springer Verlag, 1990
- [Fri-07] Friedrich, M.; Kausler, B.; Welter, J.: Modulares Tool – Konzept für den Systemtest mechatronischer Produkte. atp Automatisierungstechnische Praxis 49, Heft 9. München: Oldenbourg Industrieverlag, 2007
- [Fri-10] Friedrich, T.: Technologieorientiertes Programmier- und Steuerungssystem für Industrieroboter. Dissertation, Berlin: Technische Universität Berlin, 2010
- [Gas-07] Gasper, S.: Offline – Programmierung von Industrierobotern in automatisierten flexiblen Fertigungssystemen am Beispiel des Profilschneidens und des „Rundens beim Strangpressen“. Dissertation, Dortmund: Technische Universität Dortmund, 2007
- [Gau-07] Gauß, M.: Ein Beitrag zur Standardisierung von Kommunikationsschnittstellen für Industrieroboter und komplexe Sensoren. Dissertation, Karlsruhe: Karlsruher Institut für Technologie, 2007
- [Geo-09] Georgi, W.; Metin, E.: Einführung in LabVIEW. 4. Auflage, Leipzig: Carl Hanser Verlag, 2009
- [Gla-09] Glaser, S.; Jakobi, R.; Kraft, W. – W.; Wüst, A.: Simulieren, Herstellen, Prüfen – Komplettes Paket für die Kunststoffbauteile – Entwicklung. ATZ Automobiltechnische Zeitschrift 111, Heft 5. Wiesbaden: Vieweg Verlag, 2009

- [Göh-11] Göhner, P.: Fachpraktikum Automatisierungstechnik, Einführung in CAN. Institut für Automatisierungs- und Softwaretechnik, Stuttgart: Universität Stuttgart, 2011
- [Gös-01] Gössing, H. – P.: Virtuelle Auslegung der Karosseriestruktur für Betriebsbeanspruchungen – Entwicklung eines durchgängigen Auslegungsprozesses unter Berücksichtigung von Prozesseinflüssen aus der Fertigung. Dissertation, München: Technische Universität München, 2001
- [Got-01] Gottschald, J.: Place & Play – Roboter. Ein portables Handhabungssystem für die Werkstatt. Dissertation, Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 2001
- [Gra-06] Grabner, J.: Konstruieren von PKW – Karosserien. Berlin: Springer Verlag, 2006
- [Gre-10] Grebenstein, M.; Chalon, M.; Hirzinger, G.; Siegwart, R.: Antagonistically Driven Finger Design for the Anthropomorphic DLR Hand Arm System. HUMANOIDS, 6. – 9. 12.2010, Nashville (Tennessee, USA)
- [Gro-06] Grotepaß, J.: CheckMATE: Mobile Serviceroboter zur schritthal tenden Optimierung von Handhabung und Inspektion in der Klein- und Mittelserienfertigung. Weinheim: Freudenberg Dichtungs- und Schwingungstechnik GmbH & Co KG, 2006
- [Gus-06] Guserle, S.; Kruppa, A.; Schöttl, C.; Streicher, M.: Reduzierung der Entwicklungszeiten durch Integration der FEM – Simulation in den Freigabeprozess von Nutzfahrzeugen. VDI Berichte Nr. 1967: Berechnung und Simulation im Fahrzeugbau, Düsseldorf: VDI Verlag, 2006
- [Häg-09] Hägele, M.: Final Project Workshop: The European Robot Initiative for Strengthening the Competitiveness of SMEs in Manufacturing. Stuttgart, 2009
- [Hai-06] Haibach, E.: Betriebsfestigkeit. Verfahren und Daten zur Bauteilberechnung. 3. Auflage, Berlin: Springer Verlag, 2006
- [Hau-04] Hauke, M.: Simulation des Missbrauchverhaltens von Gesamtfahrzeugen. Dissertation, München: Technische Universität München, 2004
- [Hau-07] Haun, M.: Handbuch Robotik. Programmieren und Einsatz intelligenter Roboter. Berlin: Springer Verlag 2007
- [Hei-07] Hein, B.; Hensel, M.: Modellgestützte Online – Programmierverfahren für Industrieroboter. 40. Sitzung des FA 4.13 „Steuerung und Regelung von Robotern“. Karlsruhe: Universität Karlsruhe, 2007
- [Hel-06] Helms, E.: Roboterbasierte Bahnführungsunterstützung von industriellen Handhabungs- und Bearbeitungsprozessen. Dissertation, Stuttgart: Universität Stuttgart: 2006
- [Hen-09] Henrich, D.; Deiterding, J.: Spezialisten überflüssig – Die Sensorik Integration. Computer & Automation, 10.2009. Kissing: WEKA Media, 2009

- [Her-07] Herfeld, U.: Matrix – basierte Verknüpfung von Komponenten und Funktionen zur Integration von Konstruktion und numerischer Simulation. Dissertation, München: Technische Universität München, 2007
- [Her-08] Herkommer, G.: Neue Programmierkonzepte für Roboter: Schneller Lernen – feinfühler bei der Arbeit. Computer & Automation, 26.05.2008. Kissing: WEKA Media, 2008
- [Her-09a] Herkommer G.: Das Potential moderner SPS – Technik. Computer & Automation, 16.06.2009. Kissing: WEKA Media, 2009
- [Her-09b] Herkommer, G.: Das Potential des Microsoft Robotics Developer Studio. Computer & Automation, 14.04.2009. Kissing: WEKA Media, 2009
- [Her-10] Herkommer, G.: Kollege Roboter zeigt Feingefühl. Computer & Automation, 17.02.2010. Kissing: WEKA Media, 2010
- [Hil-07] Hilfrich, E.; Maas, J.; Flöth, T.; Türen – Benchmark. Vergleich und Optimierung von Fahrzeugtüren. ATZ Automobiltechnische Zeitschrift 109, Heft 6. Wiesbaden: Vieweg Verlag, 2007
- [Hof-98] Hoffmann, R.: Signalanalyse und Erkennung. Eine Einführung für Informationstechniker. Berlin: Springer Verlag, 1998
- [Hof-06] Hoffmann, R.: Analyse und Restbussimulation für FlexRay-Netzwerke. Weingarten: IXXAT Automation GmbH, 2006
- [Hol-95] Hollenberg, F.: CAD – basierte Offline – Programmierung von Lichtbogenschweißrobotern. Dissertation, Aachen: Rheinisch-Westfälische Technische Hochschule Aachen, 1995
- [Hol-06] Holzäpfel, M.: Entwicklung eines Teach – Systems für Industrieroboter auf der Basis von Orientierungssensoren. Bachelorarbeit, Reutlingen: Hochschule Reutlingen, 2006
- [Hue-10] Huelke, M.; Umbreit, M.; Ottersbach, H. J.: Sichere Zusammenarbeit von Mensch und Industrieroboter. MM Maschinenmarkt 33. Würzburg: Vogel Business Media, 2010
- [Hül-09] Hülsebusch, D.; Ungethüm, J.; Braig, T.; Dittus, H.: Multidisziplinäre Simulation von Fahrzeugen. ATZ Automobiltechnische Zeitschrift 111, Heft 10. Wiesbaden: Vieweg Verlag, 2009
- [Hum-98] Humburger, R.: Konzeption eines Systems zur aufgabenorientierten Roboterprogrammierung. Dissertation, Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 1998
- [Iav-08] Karosserieversuch - gerüstet für alle Prüfaufgaben. Produktdatenblatt zu Prüfeinrichtungen im Karosserieversuch. IAV GmbH, 2008
- [Ihm-07] Ihme, T.: Robotik. Vorlesungsscript. Mannheim: Hochschule Mannheim, 2007



- [Ima-10] ImagingLab: Robotics Library for DENSO, KUKA and Mitsubishi. Toolkit for Directly Commanding and Controlling Robotics from NI LabVIEW. Lodi (Italy), 2010
- [Jas-05] Jasperneite, J.: Echtzeit Ethernet im Überblick. atp Automatisierungstechnische Praxis 47, Heft 3. München: Oldenbourg Industrieverlag, 2005
- [Jas-08] Jaschke, J.: Die modulare SPS – Software. Computer – Automation, 02.10.2008. Kissing: WEKA Media, 2008
- [Joo-07] Joos, M.: Neukonstruktion einer flexiblen Greifhand. Diplomarbeit, Sigmaringen: Hochschule Albstadt – Sigmaringen, 2007
- [Jun-04] Jungmann, T.: Der virtuelle Test ist in vielen Fällen bereits zur Realität geworden. Interview mit Dr. Rainer, 21.01.2004. ATZonline, Wiesbaden: Springer Automotive Media, 2004
- [Kai-09] Kaiser, L.; Nordsiek, D.; Terfloth, A.: Softwaregestützte Konzipierung mechatronischer Systeme und der zugehörigen Produktionssysteme. ATZelektronik 4, Heft 5. Wiesbaden: Vieweg Verlag, 2009
- [Kam-09] Kammerer, C.; Schmidt, R.; Hochmann, G.: Der Rollenprüfstand als Entwicklungsplattform. ATZ Automobiltechnische Zeitschrift 111, Heft 11. Wiesbaden: Vieweg Verlag, 2009
- [Kaw-08] Produktdatenblatt Kawasaki Robot FS20N. Neuss: Kawasaki Robotics GmbH, 2008
- [Kba-10] Kraftfahrt – Bundesamt: Jahresbericht 2010. Flensburg: Kraftfahrt – Bundesamt, 2010
- [Kis-07] Betriebsanleitung Ladungsmeter Typ 5015A. Winterthur (Schweiz): Kistler Holding AG, 2007
- [Kis-10a] Produktdatenblatt Quarz – Kraftmesselemente Typ 9301B ... 9371B. Winterthur (Schweiz): Kistler Holding AG, 2010
- [Kis-10b] Produktdatenblatt 3 – Komponenten – Kraftmesselement Typ 9327C. Winterthur (Schweiz): Kistler Holding AG, 2010
- [Kle-99] Kleinmichel, G.; Truöl, C.: Hierarchisches Steuerungskonzept für die Prüfautomatisierung. Dresden: ADG Automatisierung Dresden GmbH, 1999
- [Kne-04] Kneupner, Klaus: Entwicklung eines Programmier- und Steuerungskonzepts für Robotersysteme auf der Basis eines Umweltmodells. Dissertation, Dortmund: Technische Universität Dortmund, 2004
- [Kol-08] Kollmer, H.; Janßen, A.; Küçükay, F.: Kundennahe Fahrzeugmessungen. dSpace news 1/2008. Paderborn, 2008

- [Köp-01] Köpf, P.; Bürgel, E.: Wandel der Mess- und Versuchstechnik in der Systemkette zwischen OEM und Entwicklungspartnern. VDI Berichte Nr. 1616: Mess- und Versuchstechnik im Fahrzeugbau, Düsseldorf: VDI Verlag, 2001
- [Kor-09] Korotkiy, D.; Bender, K.; Treffny, R.: Entwicklung eines Testsystems in der Praxis. Leitfaden zur Anwendung der universellen Testsystem – Architektur. Manuskript im Rahmen des FuE – Projekts TeSiM, 2009
- [Kor-10] Korotkiy, D.: Universelle Testsystem – Architektur in der Mechatronik. Dissertation, München: Technische Universität München, 2010
- [Kra-09] Kramer, F.: Passive Sicherheit von Kraftfahrzeugen. Biomechanik – Simulation – Sicherheit im Entwicklungsprozess. ATZ / MTZ Fachbuch. Wiesbaden: Vieweg Verlag, 2009
- [Küb-10] Produktdatenblatt Inkrementale Drehgeber Sendix 5000 / 5020. Villingen – Schwenningen: Fritz Kübler GmbH, 2010
- [Kug-99] Kugelman, D.: Aufgabenorientierte Offline – Programmierung von Industrierobotern. Dissertation, München: Technische Universität München, 1999
- [Kuk-09] Produktdatenblatt KUKA KR 16 – 2 S. Gersthofen: KUKA Roboter GmbH, 2009
- [Kun-05] Kunkel, R.; Schammer, K.; Ischen, P.; Bathelt, H.: Kombiniertes Fahrzeug- und Aggregateprüfstand für die Akustikentwicklung. ATZ Automobiltechnische Zeitschrift 107, Heft 4. Wiesbaden: Vieweg Verlag, 2005
- [Kup-09] Kuppinger, S.: Schrittketten – Programmierung für Messtechniker. Computer – Automation, 11.08.2009. Kissing: WEKA Media, 2009
- [Lau-04] Lauzi, M.: IMAC – Integrated Modular Automation Concept. atp Automatisierungstechnische Praxis 46, Heft 2. München: Oldenbourg Industrieverlag, 2004
- [Le-09] Le, H.; Friedl, N.; Ramsbrock, J.; Martinus, M.: Fokussierung auf den Testprozess – Methoden zur Testfall – Priorisierung. ATZelektronik 4, Heft 3. Wiesbaden: Vieweg Verlag, 2009
- [Lec-06] Lechner, H.: Homologation basierend auf virtuellen Tests. VDI Berichte Nr. 1967: Berechnung und Simulation im Fahrzeugbau, Düsseldorf: VDI Verlag, 2006
- [Lem-11] Produktdatenblatt Current Transducer HASS 50..600-S. Plan – les – Ouates (Schweiz): LEM Holding SA, 2011
- [Leo-06] Leon, A.: Entwicklung, Aufbau und Test eines robotergeführten Fräswerkzeugs zur Bearbeitung von Holzwerkstoffen. Diplomarbeit, Stuttgart: Universität Stuttgart, 2006

- [Ler-07] Lerch, R.: Elektrische Messtechnik. Analoge, digitale und computergestützte Verfahren. 4. Auflage, Berlin: Springer Verlag, 2007
- [Let-02] Lettenbauer, H.: System zum Einsatz von Industrierobotern in der fertigungsnahen Qualitätsprüfung. Dissertation, Hamburg: Universität der Bundeswehr Hamburg, 2002
- [Lie-91] Liebenow, D.: Ein Beitrag zur Makroprogrammierung in der automatisierten schweißtechnischen Fertigung. Dissertation, Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 1991
- [Lie-04] Lieven, W.; Warnecke, U.; Tran – Quoc, T.: Virtuelle Dauerlaufauslegung von PKW – Karosserien. VDI Berichte Nr. 1846: Berechnung und Simulation im Fahrzeugbau, Düsseldorf: VDI Verlag, 2004
- [Lin-08] Lindermaier, R.: Das Türsystem BMW 7er. ATZ Extra 11 / 2008. Wiesbaden: Vieweg Verlag, 2008
- [Lin-09] Linnemann, H.: Begleitfolien zum Wahlpflichtfach Robotertechnik im Studiengang Technische Informatik. Berlin: Technische Fachhochschule Berlin, 2009
- [Lin-10] Lindemann, U.: Methoden der Produktentwicklung. Vorlesungsscript. München: Technische Universität München, 2010
- [Löb-06] Löber, P.; Nawaz, R.: A modular real – time coordinate transformation for a redundant 9 – axis industrial robot. VDI Berichte Nr. 1956: International Symposium on Robotics, Düsseldorf: VDI Verlag, 2006
- [Löb-07] Löber, P.; Nawaz, R.; Hesse, M.; Schindler, M.: Vorlesungsmanuskript Robotik. 4. Auflage, Freiberg: Technische Universität Bergakademie Freiberg, 2007
- [Löp-04] Löper, C.: Anforderungsanalyse und Anforderungsdefinition für sicherheitskritische Systeme. Paderborn: Universität Paderborn, 2004
- [Lüd-05] Lüdemann – Ravit, B.: Ein System zur Automatisierung der Planung und Programmierung von industriellen Roboterapplikationen. Dissertation, Dortmund: Technische Universität Dortmund, 2005
- [Mai-07] Maisch, M.: Zuverlässigkeitsorientiertes Erprobungskonzept für Nutzfahrzeuggetriebe unter Berücksichtigung von Betriebsdaten. Dissertation, Stuttgart: Universität Stuttgart, 2007
- [Mar-07] Marscholik, C.; Subke, P.: Datenkommunikation im Automobil – Grundlagen, Bussysteme, Protokolle und Anwendungen. Heidelberg: Hüthig Verlag, 2007
- [Mat-06] Matthias, B. et al.: Abschlussbericht POTHOS – Portable Handhabungssysteme für den ortsflexiblen Einsatz in der Produktion. Karlsruhe: Projektträger Karlsruhe, 2006
- [Mau-07] Mau, G.: Sicher ein Team – Kamergesteuerte Mensch – Roboter – Kooperation. handling, Heft 7 / 2007. Darmstadt: Hoppenstedt Verlag, 2007

- [May-06] Mayer, G.: Vorlesungsscript Numerische Mathematik. Rostock: Universität Rostock, 2006
- [Mer-06] Merkt, T.; Krastel, M.: Virtuelle Produktabsicherung auf Basis eines CAE – Datenmanagements. VDI Berichte Nr. 1967: Berechnung und Simulation im Fahrzeugbau, Düsseldorf: VDI Verlag, 2006
- [Mes-03] Mess- und Versuchstechnik in der Fahrzeugentwicklung. Kongress in Würzburg. Düsseldorf: VDI Verlag, 2003
- [Mey-06] Meyer, C.; Schäfer, T.: Intuitive Industrieroboterprogrammierung. Automation in der Holzwirtschaft. Biel (Schweiz) 12. – 13.10.2006
- [Mey-07a] Meyer, C.; Hollmann, R.; Parlitz, C.; Hägele, M.: Programmieren durch Vormachen für Assistenzsysteme – Schweiß und Klebebahnen intuitiv programmieren. it Information Technology 49, Heft 4. München: Oldenbourg Wissenschaftsverlag, 2007
- [Mey-07b] Meywerk, M.: CAE – Methoden in der Fahrzeugtechnik. Berlin: Springer Verlag, 2007
- [Mit-10] Produktdatenblatt Mitsubishi MELFA Industrieroboter RV – 12SDL. Ratingen: Mitsubishi Electric Europe B.V., 2010
- [Mor-09] Morphee2 – Automation System. Produktbeschreibung. Darmstadt: D2T, 2009
- [Mot-08] Produktdatenblatt Industrieroboter Motoman – HP20. Allershausen: Motoman, 2008
- [Mot-10] Produktdatenblatt Industrieroboter Motoman – SDA10D. Allershausen: Motoman, 2010
- [Mrk-10] SafeInteraction und SafeGuiding. Produktbeschreibungen der Firma MRK - Systeme GmbH. Augsburg, 2010
- [Mül-98] Müller, M.: Process an Device for Teaching a Program – Controlled Robot. WIPO Patent Application WO/1998/000766. Assignee: KUKA Schweißanlagen GmbH, Augsburg, 1998
- [Mül-06] Müller, C.; Hartmann, N.; Sax, E.; Holzenkamp, C.: Automatisiertes Testen von Steuergeräten der Dachbedieneinheit. ATZ Automobiltechnische Zeitschrift 108, Heft 11. Wiesbaden: Vieweg Verlag, 2006
- [Müt-05] Mütterich, H.: Konzeption und Entwicklung von robotergeführten Werkzeugsystemen zur Fertigung von Holzbauteilen in kleinen Losgrößen. Diplomarbeit, Dortmund: Technische Universität Dortmund, 2005
- [Müt-09] Mütterlein, B.: Handbuch für die Programmierung mit LabVIEW. Heidelberg: Spektrum Akademischer Verlag, 2009
- [Nac-08] Produktdatenblatt Nachi Presto MC Series Robot MC20-01. Novi (Michigan, USA): Nachi Robotic Systems Inc., 2008

- [Nat-06] National Instruments: Designing Next Generation Test Systems. An In – depth Developers Guide. 2006
- [Nat-08] Produktdatenblatt DAQ NI 622x, NI 625x, and NI 628x Devices. Austin (Texas, USA): National Instruments Corporate Headquarters, 2008
- [Nat-09a] National Instruments: Entwurf automatisierter Prüfsysteme. Ein praktischer Leitfaden für softwaredefinierte Prüfsysteme. 2009
- [Nat-09b] National Instruments: Grundlagen softwaredefinierter Prüfsysteme. Die Architektur modularer, leistungstarker Prüfsysteme verstehen lernen. 2009
- [Nat-09c] Produktdatenblatt DAQ NI 660x. Austin (Texas, USA): National Instruments Corporate Headquarters, 2009
- [Nat-10a] National Instruments: Entwicklung von Prüfständen der nächsten Generation. 2010
- [Nat-10b] National Instruments: Developing a modular software – architecture. 2010
- [Nau-07] Naumann, M.; Wegener, K.; Schraft, R.: Control Architecture for Robot Cells to Enable Plug’n’Produce. IEEE International Conference on Robotics and Automation. Roma (Italy) 10. – 14.04.2007
- [Naw-08] Nawotki, A.; Rohrbach, J.: CAE Datenmanagement – ein Praxisbericht. VDI Berichte Nr. 2031: Berechnung und Simulation im Fahrzeugbau, Düsseldorf: VDI Verlag, 2008
- [Neu-04] Neureiter, W., Sukarie, G.: Simulation des dynamischen Betätigungsverhaltens eines PKW – Türschlosses. VDI Berichte Nr. 1846: Berechnung und Simulation im Fahrzeugbau, Düsseldorf: VDI Verlag, 2004
- [Obe-06] Oberer, S.: LS – DYNA Simulation of Robot – Dummy Crash Tests for Robot Safety Assessment. LS – DYNA Anwenderforum. Ulm, 12. – 13.10.2006
- [Obe-07] Oberer, S.; Schraft, R.: Robot – Dummy Crash Tests for Robot Safety Assessment. IEEE International Conference on Robotics and Automation. Roma (Italy), 10. – 14.04.2007
- [Omr-08] Produktdatenblatt Cylindrical Inductive Proximity Sensor – E2A Single Sensing Distance. Hoofddorf (Niederlande): OMRON EUROPE B.V., 2008
- [Otc-09] Produktdatenblatt OTC Lichtbogen – Schweißroboter AII Serie V20. Mönchengladbach: OTC DAIHEN EUROPE GmbH, 2009
- [Pah-07] Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K.-H.: Pahl/Beitz Konstruktionslehre. Grundlagen erfolgreicher Produktentwicklung. Methoden und Anwendung. Berlin: Springer Verlag, 2007
- [Pas-02] Paschek, L.: ATZ – Konferenz: An der Simulation führt kein Weg vorbei. Interview mit Prof. em. B. Heiing, 18.09.2002. ATZonline, Wiesbaden: Springer Automotive Media, 2002

- [Pet-08] Peti, P.; Timmerberg, A.; Pfeffer, T.; Müller, S.; Rätz, C.: Automatische Validierung der Diagnoseservices. ATZelektronik 3, Heft 6. Wiesbaden: Vieweg Verlag, 2008
- [Pep-97] Peper, S.: Schweißstruktur – orientierte Offline – Programmierung von Lichtbogenschweißrobotern. Dissertation, Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 1997
- [Pie-09] Pierburg, L.: Entwicklung eines Lebensdauersensors. Dissertation, Aachen: : Rheinisch – Westfälische Technische Hochschule Aachen, 2009
- [Pir-05] Pires, N.: Robot – by – voice: Experiments on commanding an industrial robot using the human voice. University of Coimbra (Portugal), 2005
- [Pir-06] Pires, N.: Robotics for small and medium enterprises: control and programming challenges. University of Coimbra (Portugal), 2006
- [Pir-08] Pires, N.: New challenges for industrial robotic cell programming. University of Coimbra (Portugal), 2008
- [Pös-05] Pöschl, M.; Jack, P.; Kausler, B.: Grafische Testfallspezifikation für den Applikationstest mechatronischer Produkte. atp Automatisierungstechnische Praxis 47, Heft 8. München: Oldenbourg Industieverlag, 2005
- [Pos-04] Posch, T.; Birken, K.; Gerdorf, M.: Basiswissen Softwarearchitektur. Heidelberg: dpunkt Verlag, 2004
- [Prö-10] Pröpper, U.; Schönherr, C.: Psychoakustisches Messverfahren für den Klangeindruck von Pkw – Türen. ATZ Automobiltechnische Zeitschrift 112, Heft 6. Wiesbaden: Vieweg Verlag, 2010
- [Rad-07] Radaj, D.; Vormwald, M.: Ermüdungsfestigkeit. Grundlagen für Ingenieure. 3. Auflage, Berlin: Springer Verlag, 2007
- [Rag-05] Raguse, K.; Derpmann – Hagenström, P.; Köller, P.; Wolf, A.: Abgleich von Versuch und Simulation beim Crashtest anhand präzise überlagerter Visualisierung von Film und Berechnungsanimation, VDI Berichte Nr. 1900: Erprobung und Simulation, Düsseldorf: VDI Verlag, 2005
- [Rau-09] Rauh, S.: Automatische Prüfprogrammerstellung beim Netzwerktest. 7. Technologietag „Prüfstandskonzepte in der Automobilindustrie“. National Instruments, 2009
- [Rec-05] Recknagel, M.: Integriertes Qualitätsinformations- und Rechensystem für die dokumentierte Prüfung von Bauteilen. Dissertation, Stuttgart: Universität Stuttgart, 2005
- [Rei-06] Produktdatenblatt Reis RV20 – 16. Obernburg: Reis GmbH & Co. KG, 2006
- [Rei-08] Reif, K.: Automobilelektronik. Eine Einführung für Ingenieure. ATZ / MTZ Fachbuch. 3. Auflage, Wiesbaden: Vieweg Verlag, 2008



- [Rei-09] Reinhart, G.; Hatwig, J.; Kronthaler, M.; Musiol, J.: Offline Nachteachen von Industrierobotern als Werkzeug in einer Simulationsumgebung. München: Technische Universität München, 2009
- [Reu-06] Reussner, R.; Hasselbring, W.: Handbuch der Software – Architektur. Heidelberg: dpunkt Verlag, 2006
- [Reu-08] Reuss, H. – C.; Baumann, G.; Brost, M.: Automatisierter Test zustandsbasierter Systeme im Kraftfahrzeug. ATZelektronik 3, Heft 6. Wiesbaden: Vieweg Verlag, 2008
- [Ric-04] Richling, J.: Eigenschaften mobiler und eingebetteter Systeme: Weiche Echtzeit. Vorlesungsscript. Berlin: Humboldt – Universität zu Berlin, 2004
- [Rie-05] Riener, H.; Mayr, A.; Steinbatz, M.: Verbesserungspotentiale bei Zeit und Kosten durch Integration des „virtuellen Prüfstands“ in den Simulations- und Prüfprozess. Magna Steyr, 2005
- [Rie-07] Riegraf, T.; Beeh, S.; Krauß, S.: Effizientes Testen in der Automobilelektronik – Von der Simulation bis zur Diagnose. ATZ Automobiltechnische Zeitschrift 109, Heft 7-8. Wiesbaden: Vieweg Verlag, 2007
- [Ros-04] Rosenstein, B.; Mintgen, R.; Brehler, H.; Henkel, W.: Stufenloser Türfeststeller. ATZ Automobiltechnische Zeitschrift 106, Heft 3. Wiesbaden: Vieweg Verlag, 2004
- [Rös-09] Rösler, F.; Battenberg G.; Schüttler, F.: Subjektive Empfindungen und objektive Charakteristika von Bedienelementen. ATZ Automobiltechnische Zeitschrift 111, Heft 4. Wiesbaden: Vieweg Verlag, 2009
- [San-08] Sander, M.: Sicherheit und Betriebsfestigkeit von Maschinen und Anlagen. Konzepte und Methoden zur Lebensdauervorhersage. Berlin: Springer Verlag, 2008
- [Schä-07] Schäfer, T.: Verfahren zur hämmernden Blechumformung mit Industrieroboter. Dissertation, Stuttgart: Universität Stuttgart, 2007
- [Schi-01] Schilling, K.; Roth, H.; Rösch, O.: Fernsteuerung und Telesensorik für mobile Roboter in Raumfahrt, Industrie und Ausbildung. AT Automatisierungstechnik 49, Heft 8. München: Oldenbourg Wissenschaftsverlag, 2001
- [Schl-09] Schleißheimer, H. – J.: Applikationsnahe Restbussimulation. automotive 1-2.2009. München: Carl Hanser Verlag, 2009
- [Schm-07] Schmid, C.; Diedrich, C.: Konzept einer individuellen Fahrzeugprüfung mit formalisierten Modellen und Methoden. atp Automatisierungstechnische Praxis 49, Heft 6. München: Oldenbourg Industrieverlag, 2007
- [Schö-09] Schönitz, T.: CAN – Logdatei – TDM – Konverter für NI Diadem. 7. Technologietag „Prüfstandskonzepte in der Automobilindustrie“. National Instruments, 2009

- [Schö-11] Schönherr, S.: Druckluft ist teuerste Energie. Deutsche Handwerks Zeitung, Ausgabe 6/2011. Bad Wörlshofen: Holzmann Verlag, 2011
- [Schr-03] Schraft, R. D.; Helms, E.: rob@work – Assistenzroboter als Helfer in der Produktion. atp Automatisierungstechnische Praxis 45, Heft 2. München: Oldenbourg Industrieverlag, 2003
- [Schr-08] Schröter, B.: Inertiale Positionserfassung zur Programmierung robotergestützter Handhabungsaufgaben. Dissertation, Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 2008
- [Schü-10] Schüssler, P.; Heim, G.: Mit Noblesse in den Fond – Das Türkonzept des Opel Meriva. ATZ Automobiltechnische Zeitschrift 112, Heft 5. Wiesbaden: Vieweg Verlag, 2010
- [Schü-11] Schüller, A.; Trottenberg, U.; Wienands, R.: Schnelle Lösung großer Gleichungssysteme. Vorlesungsscript. Köln: Universität zu Köln, 2011
- [Sda-05] Sdahl, M.; Kuhlenkötter, B.; Brutscheck, T.: Schnelle Offline-Programmierung komplexer Prozesse. Anforderungsgerechte Programmiersysteme entlasten den Bediener. VDI – Z 147, Nr. 11/12. Berlin: Springer VDI Verlag, 2005
- [Sei-08] Seiffert, U.; Rainer, G.: Virtuelle Produktentstehung für Fahrzeug und Antrieb im Kraftfahrzeug. ATZ / MTZ Fachbuch. Wiesbaden: Vieweg Verlag, 2008
- [Sha-07] Shah, J.; Gijbels, M.: Innovativer Prüfstand für Lenksysteme. ATZ Automobiltechnische Zeitschrift 109, Heft 6. Wiesbaden: Vieweg Verlag, 2007
- [Sic-10] Produktdatenblatt Industrial Safety Systems. Düsseldorf: SICK Vertriebs – GmbH, 2010
- [Sie-01] Siemon, E.: Über den Entwurf von Benutzungsschnittstellen technischer Anwendungen mit visuellen Spezifikationsmethoden und Werkzeugen. Dissertation, Darmstadt: Technische Universität Darmstadt, 2001
- [Sie-04] Siedersleben, J.: Moderne Software – Architektur. Heidelberg: dpunkt Verlag, 2004
- [Sie-06] Simon, A.: Qualitative und quantitative Analyse der linearen und nichtlinearen Schadensakkumulationshypothesen unter Einbeziehung der statistischen Versuchsplanung. Dissertation, Kassel: Universität Kassel, 2006
- [Sig-01] Sigwart, A.; Bockmühl, P.: Ein neues Testverfahren zur Verifikation der Fahrzeugelektrik/ - elektronik. VDI Berichte Nr. 1616: Mess- und Versuchstechnik im Fahrzeugbau, Düsseldorf: VDI Verlag, 2001
- [Sim-08] Simon, R.: Software – Abstraktionsschicht für Robotersteuerung. Teilgebiet: Ansteuerung. Diplomarbeit, Ingolstadt: Fachhochschule Ingolstadt, 2008
- [Sta-97] Stasch, E.: Ethernet: Ausarbeitung zum Proseminar IBM – PC. Chemnitz: Technische Universität Chemnitz, 1997

- [Sta-08] Starke, G.: Effektive Software – Architekturen: Ein praktischer Leitfaden. 3. Auflage, Leipzig: Carl Hanser Verlag, 2008
- [Sta-09] Stark, A.: Workshop: NI – Software für Messung, Kalibrierung und Diagnose an Steuergeräten. 7. Technologietag „Prüfstandskonzepte in der Automobilindustrie“. National Instruments, 2009
- [Stä-09] Schulungsunterlagen Stäubli VAL3 – Programmierschulung V1.15. Bayreuth: Stäubli Tec – Systems GmbH Robotics, 2009
- [Stä-11a] Produktdatenblatt Stäubli Industrieroboter Baureihe RX160. Bayreuth: Stäubli Tec – Systems GmbH Robotics, 2011
- [Stä-11b] Produktdatenblatt Stäubli Industrieroboter Baureihe TX60. Bayreuth: Stäubli Tec – Systems GmbH Robotics, 2011
- [Ste-06] Steinmüller, J.: Informationen zum Thema Serviceroboter. Chemnitz: Technische Universität Chemnitz, 2006
- [Ste-09] Stein, C.: Leitsystemkonzepte für die Modernisierung von Großprüfständen. 7. Technologietag „Prüfstandskonzepte in der Automobilindustrie“. National Instruments, 2009
- [Sto-05] Stopp, A.; Baldauf, T.; Horstmann, S.; Kristensen, S.: Ein Sicherheitskonzept für Roboterassistenten in der Fertigung. atp Automatisierungstechnische Praxis 47, Heft 2. München: Oldenbourg Industrieverlag, 2005
- [Str-80] Strobel, W.: Die moderne Automobilkarosserie. Stuttgart: Franckh, 1980
- [Str-02] Streilein, T.; Hillmann, J.: Stochastische Simulation und Optimierung am Beispiel VW Phaeton. VDI Tagung: Berechnung und Simulation im Fahrzeugbau, Würzburg, 2002
- [Tän-07] Tänzer, C.: Experimentelle Untersuchungen zur Erkennung der frühzeitigen Ermüdung an Kraftfahrzeugtüren. Diplomarbeit, Zwickau: Westsächsische Hochschule Zwickau, 2007
- [Ter-07] Terhart, A.: Modulare Maßanfertigungen in der Prüftechnik. ATZelextronik 2, Heft 2. Wiesbaden: Vieweg Verlag, 2007
- [Thi-03] Thiernemann, S.; Schraft, R. D.: team@work – Mensch – Roboter – Kooperation in der Montage. atp Automatisierungstechnische Praxis 45, Heft 11. München: Oldenbourg Industrieverlag, 2003
- [Thi-05] Thiernemann, S.: Direkte Mensch – Roboter – Kooperation in der Kleinteilemontage mit einem SCARA – Roboter. Dissertation, Stuttgart: Universität Stuttgart, 2005
- [Tho-02] Thomas, U.; Wahl, F. M.: Robot Programming – From Simple Moves to Complex Robot Tasks. Braunschweig: Technische Universität Braunschweig, 2002

- [Tho-08] Thomas, U.: Automatisierte Programmierung von Robotern für Montageaufgaben. Dissertation, Braunschweig: Technische Universität Braunschweig, 2008
- [Tim-11] Timmerman, M. et al.: Comp.realtime: Frequently Asked Questions V3.5.: <http://www.faqs.org/faqs/realtime-computing/faq>. Abgerufen am 18.06.2011
- [Tru-06a] Truöl, C.: Prüfstandstechnik für mechatronische Baugruppen. Struktureller Aufbau, Funktionalität, Erfahrungen. Ölhydraulik und Pneumatik 4/2006. Mainz: Vereinigte Fachverlage, 2006
- [Tru-06b] Truöl, C.: Test und Diagnose mechatronischer Baugruppen während des Fertigungsprozesses. Comtec 2006. Dresden: ADG Automatisierung Dresden GmbH, 2006
- [Tru-10] Truöl, C.: LabVIEW RT prüft mechatronische Baugruppen - Prüfstandskonzept, Funktionalität, Erfahrungen. National Instruments, 2010
- [Ung-09] Ungermann, J.: Zuverlässigkeitsnachweis und Zuverlässigkeitsentwicklung in der Gesamtfahrzeugerprobung. Dissertation, Zürich: Eidgenössische Technische Hochschule Zürich, 2009
- [Vah-08] Vahling, K.: Software – Abstraktionsschicht für eine Robotersteuerung. Teilgebiet: Anlernvorgang. Diplomarbeit, Bocholt: Fachhochschule Gelsenkirchen, Abt. Bocholt, 2008
- [Vec-10] Vector CANoe FDX Protocol Version 1.0.1. Stuttgart: Vector Informatik GmbH, 2010
- [Vec-11a] Online – Hilfe der Software Vector CANoe V7.5.6.6. Stuttgart: Vector Informatik GmbH, 2011
- [Vec-11b] Internetseite der Firma Vector Informatik, Produkte für Steuergeräte – Tests: CANoe 7.6: [http://www.vector.com/vi\\_canoe\\_de.html](http://www.vector.com/vi_canoe_de.html). Abgerufen am 15.04.2010
- [Vit-04] Vitturi, S.: PC – based automation systems: an example of application for the real – time control of blowing machines. Computer Standards & Interfaces 26. Amsterdam: Elsevier Verlag, 2004
- [Völ-03] Völkel, A.: Evolutionäre Optimierung von grafischen Bedienoberflächen zur Führung technischer Prozesse. atp Automatisierungstechnische Praxis 45, Heft 9. München: Oldenbourg Industrieverlag, 2003
- [Vmo-12] Internetseite des Bundesministerium des Innern: Das V-Modell XT: <http://www.v-modell-xt.de>. Abgerufen am 21.01.2012
- [Vog-99] Vogel, J.; Brunner, B.; Landzettl, K.; Hirzinger, G.: Internet Virtual Reality Technologien zur Fernvisualisierung für Teleservice- und Telerobotikanwendungen. GMA – Fachtagung: Industrielle Automation und Internet / Intranet – Technologien, 08. / 09.11.1999. Langen: 1999

- [Vog-05] Vogel, O.; Arnold, I.; Chughtai, A.; Völter, M.: Software - Architektur. Grundlagen - Konzepte – Praxis. Heidelberg: Spektrum Akademischer Verlag, 2005
- [Vog-07a] Vogelpohl, J.: Automatisierung eines Türdauerlaufs mittels LabVIEW unter Zuhilfenahme der in Fahrzeug verbauten Elektronik und eines Knickarmroboters. Diplomarbeit, Osnabrück: Fachhochschule Osnabrück, 2007
- [Vog-07b] Vogelsang & Benning: Produktdatenblatt Lebensdauerprüfung von Fensterhebermodulen. Bochum: Vogelsang & Benning, 2007
- [Vog-08] Vogl, W.: Eine interaktive räumliche Benutzerschnittstelle für die Programmierung von Industrierobotern. Dissertation, München: Technische Universität München, 2008
- [Vog-10] Vogelpohl, J.: Exakter Dauerläufer – Robotik in der Mess- und Versuchstechnik. Handling, Heft 5 /2010. Darmstadt: Hoppenstedt Verlag, 2010
- [Vog-11a] Vogelpohl, J.: Reale Dauererprobung – Prüfstand für bewegliche Karosseriekomponenten. Bertrandt Magazin 09 / 2011. Ehningen: Bertrandt AG, 2011
- [Vog-11b] Vogelpohl, J.: Robotik in der Komponentenerprobung. OEM Supplier 2011. München: Carl Hanser Verlag, 2011
- [Voß-08] Voß, R.: Bewertung des Einflusses von Randbedingungen für eine Toleranzsimulation nachgiebiger Baugruppen im Karosserierohbau. 19. Symposium „Design for X“. Neukirchen, 9. – 10.10.2008
- [Wag-03] Wagner, T.; Göhner, P.; Urbano, P.: Softwareagenten – Einführung und Überblick über eine alternative Art der Softwareentwicklung. atp Automatisierungstechnische Praxis 45, Heft 10. München: Oldenbourg Industrieverlag, 2003
- [Wal-02] Walser, H.: Affine Abbildungen. Skript - Modul zur Vorlesung Geometrie. Zürich: Zürich: Eidgenössische Technische Hochschule Zürich, 2002
- [Wal-06] Wallentowitz, H.; Reif, K.: Handbuch Kraftfahrzeugelektronik. ATZ / MTZ Fachbuch. Wiesbaden: Vieweg Verlag, 2006
- [Wan-04] Wanoschek, S.; Rasche, R.; Subke, P.: Automatische Steuergeräte – Funktionsprüfung. auto & elektronik 2 / 2004. Heidelberg: Hüthig Verlag, 2004
- [War-04] Wartack, S.: Prüfstand für den Tür – Seitencrash. Automotive Engineering Partners 3/2004. Wiesbaden: Vieweg Verlag, 2004
- [Web-09] Weber, W.: Industrieroboter – Methoden der Steuerung und Regelung. Leipzig: Carl Hanser Verlag, 2009
- [Wec-06] Weck, M.; Brecher, C.; Almeida, C.: Unscharfe Roboterprogrammierung – Programmierung und Durchführung sensorgestützter Handhabungsaufgaben. Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 2006

- [Wes-04] Westhoff, D.; Zhang, J.; Stanek, H.; Scherer, T.; Knoll, A.: Mobile Manipulatoren und ihre aufgabenorientierte Programmierung. atp Automatisierungstechnische Praxis 46, Heft 10. München: Oldenbourg Industrieverlag, 2004
- [Wes-09] Westkämper, E.; Verl, A.: Fraunhofer – Institut für Produktionstechnik und Automatisierung. Jahresbericht. Stuttgart, 2009
- [Wie-99] Wiegandt, P.: Kundenspezifische Prüfstände für Magnetventile. Dresden: ADG Automatisierung Dresden GmbH, 1999
- [Wie-05] Wienke, L.; Müller, A.: Softwareunterstützte Messdatenauswertung zur Optimierung von Prozessen im Baustoffrecycling. Weimar: Bauhaus – Universität Weimar, 2005
- [Wie-08] Wiehe, A.; Kern, S.; Maas, J.: Rotatorischer MRF – Aktor für einen Türassistenten. AT Automatisierungstechnik 56, Heft 3. München: Oldenbourg Wissenschaftsverlag, 2008
- [Wey-11] Weyer, D.: Entwicklung von Roboterwerkzeugen für die Lebensdauerprüfung von Fahrzeugtüren. Bachelorarbeit, Nürnberg: Georg – Simon – Ohm – Hochschule Nürnberg, 2010
- [Wic-07] Wichmann, K.: Auswertung von Messdaten. München: Oldenbourg Verlag, 2007
- [Win-06] Winkler, A.: Ein Beitrag zur kraftbasierten Mensch – Roboter – Interaktion. Dissertation, Chemnitz: Technische Universität Chemnitz, 2006
- [Win-08] Winkler, U.: Roboter „erlebt“ die Qualität. Automobil – Produktion 04 / 2008, Landsberg: Verlag moderne Industrie GmbH, 2008
- [Wör-03] Wörn, H.: Tendenzen in der Fabrikautomation und Robotik. atp Automatisierungstechnische Praxis 45, Heft 7. München: Oldenbourg Industrieverlag, 2003
- [Woi-11] Woithe, A.: Konstruktion einer transportablen Anbindung für Roboter an Kraftfahrzeugen. 2. Bericht zum Praxisseminar II. München: Bertrand AG, 2011
- [Wol-09] Wolf, J.: Steuerungsintegrierte, adaptive Programmausführung einer aufgabenorientierten Programmierung in STEP – NC. Dissertation, Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 2009
- [Wul-03] Wulfsberg, J.P.; Ludwig, R.; Wessel, O.: Robotergeführte Inline – Messtechnik. VDI – Z 145, Nr. 7/8. Berlin: Springer VDI Verlag, 2003
- [Wul-04] Wulfsberg, J. P.; Ludwig, R.: Robotergestützte Qualitätskontrolle mit intelligenter Sensorik. VDI Berichte Nr. 1841: Robotik, Düsseldorf: VDI Verlag, 2004



- [Wul-05] Wulfsberg, J. P.; Derfling, D.: Einsatz von Industrierobotern als integrierte Fertigungs- und Koordinatenmessmaschinen. Hamburg: Helmut – Schmidt – Universität – Universität der Bundeswehr Hamburg, 2005
- [Wul-09] Wulfsberg, J.P.; Clausing, N.: Ortsflexibler Industrierobotereinsatz. Hamburg: Helmut – Schmidt – Universität – Universität der Bundeswehr Hamburg, 2009
- [Zab-93] Zabel, A.: Werkstattorientierte Programmierung von Industrierobotern für automatisiertes Lichtbogenschweißen. Dissertation, Aachen: Rheinisch – Westfälische Technische Hochschule Aachen, 1993
- [Zäh-05] Zäh, M.; Vogl, W.; Munzert, U.; Spitzweg, M.: Augmented Reality – Benutzerschnittstelle für Industrieroboter. atp Automatisierungstechnische Praxis 47, Heft 11. München: Oldenbourg Industrieverlag, 2005
- [Zim-10] Zimmermann, W.; Schmidgall, R.: Bussysteme in der Fahrzeugtechnik – Protokolle und Standards. ATZ / MTZ Fachbuch, 4. Auflage. Wiesbaden: Vieweg Verlag, 2010
- [Züh-04] Zühlke, D.: Useware – Engineering für technische Systeme. Berlin: Springer Verlag, 2004
- [Zwi-08] Zwick / Roell: Prüfmaschinen und Prüfsysteme für die Automobilbranche. Ulm, 2008